Security and Privacy in Emerging Communication Standards

Haibat Khan

Thesis submitted to the University of London
for the degree of Doctor of Philosophy

Information Security Group
Royal Holloway, University of London

2020

Declaration

These doctoral studies were conducted under the supervision of Prof. Keith M. Martin.

The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, whilst enrolled in the Information Security Group as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Haibat Khan
March, 2020

Dedicated to the heroics and sacrifices of Chelsea Manning, Edward J. Snowden and Julian Paul Assange

Acknowledgments

It is not unusual, especially in the field of scientific research, to be under the misapprehension that one's accomplishments are one's alone. Nothing can be further from the truth. Once I look back, I realize that I was fortunate to cross paths with many wonderful people who helped me in many ways in pursuit of my PhD. The past three years have been thus far the most exciting, challenging, interesting, and rewarding part of my life. During these years, I have been encouraged, sustained, inspired and tolerated by a whole bunch of people. I would like to express my utmost gratitude to all of them.

I would like to begin with my deepest appreciation for my supervisor Prof. Keith Martin; for agreeing to supervise me, for his valuable guidance in how to conduct research and how to write scientific articles, for his reminders to stay focused, for his consistent encouragement and for his patience throughout this period. Thank you Keith for all your help and unconditional support.

During the course of my PhD, I had the great fortune to work with a few incredible people. In particular, I would like to thank Benjamin Dowling who introduced me to the world of formal security models and with whom I collaborated on a couple of publications. I would like to express my utmost gratitude to Steve Babbage of Vodafone for answering my endless queries in the most precise manner, for the fruitful discussions and for an exposure to the real world of cryptography.

I would like to thank my colleagues at Royal Holloway for all the wonderful times we spent together. A special shout out to Greg, Ela, Jake, Ben and Joanne for many unforgettable memories. This thesis would not have come to a successful completion without the support of the ISG staff. I would like to thank Narinder and Francesco for their support related to IT services and Jenny Lee for providing the administrative support.

My deepest gratitude to my family for their continuous and unparalleled love, help and support. I am grateful to my wife Sehrish for always being there for me as a friend and for lifting my spirits when I required it the most. Without her this thesis would only have been a dream. To my boys, Salaar and Hassan, for patiently dealing with my "unavailability" during these years.

Above all, I owe it to the Almighty God for his showers of blessings, for providing me this opportunity and granting me the capability to proceed successfully.

Abstract

5G networks are the next generation of mobile telephony systems, offering faster speeds, more reliable connections and a platform for various vertical industries. However, 5G also comes with potentially enormous privacy risks. It is therefore crucial for a successful 5G future that these privacy issues be resolved at the earliest. This thesis examines the current state of subscription privacy in 5G and highlights outstanding privacy problems and viable approaches to their resolution.

One of the most pressing privacy concerns in mobile networks relates to the exposure of the subscribers' permanent identifier, known as IMSI-catching attack. Since these identifiers uniquely identify the subscribers, malicious third parties in the past have misused their exposure to physically locate and track subscribers. Although 3GPP, the defacto international body for mobile telephony standardization, has introduced a public-key based protection mechanism to counter this threat in 5G, the proposed solution is marred with various shortcomings. Keeping in view the long-term deployment timeframes of 5G, the most significant of these shortcomings is the insecurity of the proposed mechanism against a quantum adversary. This technical problem of private identification in 5G remains open in symmetric-key settings; i.e. does there exist an efficient symmetric-key solution for private identification in 5G? This thesis answers this question positively and presents an alternative private identification scheme for 5G that works within the symmetric-key domain and overcomes the other limitations of the existing 3GPP scheme.

Another potent threat to 5G privacy is that of downgrade attacks, where a fake base station forces the connection down to one of the previous generations and then exploits the existing privacy vulnerabilities. Keeping this threat in mind, this thesis also explores the feasibility of combining the symmetric private identification scheme with a recent downgrade protection proposal to come up with a 5G identification mechanism that is both quantum-secure and downgrade-resistant.

This problem of private identification within the symmetric-key domain is of interest in other application areas too. The techniques utilized for the 5G private identification scheme are further extended to address the problem of key establishment for Wireless Body Area Networks (WBANs) in a privacy-preserving manner without resorting to public-key cryptography. This is significant because the nodes in a WBAN are energy constrained and require battery-efficient security solutions. Moreover, by avoiding public-key cryptography, a quantum-secure key agreement solution with advance security properties for the WBAN standard IEEE Std 802.15.6 is achieved.

Contents

List of Figures

List of Tables

Abbreviations

2G:	2nd Generation mobile telephone system
3G:	3rd Generation mobile telephone system
3GPP:	3rd Generation Partnership Project
4G:	4th Generation mobile telephone system
5G:	5th Generation mobile telephone system
AES:	Advance Encryption Standard
AKA:	Authentication and Key Agreement
AMF:	Access and Mobility management Function
ARIB:	Association of Radio Industries and Businesses, Japan
ARPF:	Authentication credential Repository and Processing Function
ATIS:	Alliance for Telecommunications Industry Solutions, USA
AUSF:	AUthentication Server Function
CCSA:	China Communications Standards Association
CT:	Core network and Terminals
DH:	Diffie-Hellman
ETSI:	European Telecommunications Standards Institutes
GSM:	Global System for Mobile communications
GUTI:	Globally Unique Temporary Identifier
HIPPA:	Health Insurance Portability and Accountability Act
IBC:	Identity-Based Cryptography
IBE:	Identity-Based Encryption
ICT:	Information and Communications Technology
IEC:	International Electrotechnical Commission
IEEE:	the Institute of Electrical and Electronics Engineers
IMSI:	International Mobile Subscriber Identity
IoT:	Internet of Things
ISO:	International Organization for Standardization
KDF:	Key Derivation Function
LTE:	Long-Term Evolution
MAC:	Message Authentication Code
ME:	Mobile Equipment
NIST:	National Institute of Standards and Technology
PKG:	Private Key Generator
RAN:	Radio Access Network
SA:	Services and systems Aspects
SUCI:	SUbscription Concealed Identifier
SUPI:	SUbscription Permanent Identifier
TLS:	Transport Layer Security
TMSI:	Temporary Mobile Subscriber Identity
TSG:	Technical Specification Group
TSDSI:	Telecommunications Standards Development Society, India
TTA:	Telecommunications Technology Association, Korea
TTC:	Telecommunication Technology Committee, Japan
UE:	User Equipment
UICC:	Universal Integrated Circuit Card
UMTS:	Universal Mobile Telecommunications Service
USIM:	Universal Subscriber Identity Module
V2X:	Vehicle to Everything
WBAN:	Wireless Body Area Network
WG:	Working Group

Chapter 1

Introduction

This chapter provides an executive summary of the thesis. We explain the motivation for our research and describe the structure and contributions of the thesis.

1.1 Motivation

This thesis is motivated by the application of recent cryptographic research to real-world scenarios. The aim is to focus on practical research contributions, which have the potential for widespread societal implications. One of the most viable way to achieve this is to consider improvements to existing cyber security standards.

During the last few decades a broad range of standards have been developed covering many areas of cyber security. These standards have been issued by national and international standardization bodies, as well as by industry consortia. Many of these standards have been very widely adopted - for example, the ISO/IEC 27000 series of standards has become the default basis for information security management within organizations. Despite their widespread use, there is always room for improvement in these standards and a need to revise existing security standards.

Modern communication is evolving at a rapid pace, with new paradigms being introduced by emerging ICT standards such as 5G. End-user privacy in these modern communication systems is of great importance because of the envisaged hyper-connectivity and the potential of the unprecedented services (virtual reality, machine-type communication, vehicle-to-everything, IoT, etc.) being offered. Much of the emphasis within these emerging standards has been placed on provisioning of primary security guarantees such as confidentiality, integrity, authentication, etc., and rightly so. However, pertinent privacy aspects that could have made it into the specifications with relatively little effort have often been overlooked during this process. Moreover, a reliance on public-key cryptography in some of these communication

standards puts them at risk of being vulnerable to quantum cryptanalysis. In this thesis we focus on the privacy gaps along with the substitution of public-key cryptography with symmetric cryptography to avert quantum threats. We identify security and privacy gaps in two of the emerging communication standards, 5G and the IEEE Std 802.15.6 and propose improvements to address these gaps.

1.1.1 Privacy - More Than A Modern Technological Issue

> "*Privacy is not about having something to hide. Privacy is about protection, and that is who you are and what you believe in. That is who you want to become. Privacy is a self right. Privacy is what gives you the ability to share with the world who you are on your own terms.*"

The above statement, made back in 2016 by Edward Snowden [127], presents one of the most contentious issues of the today's modern society - privacy. In this digital era, privacy has become even more important as it is not just another technological question but, many would argue, a fundamental human right. Privacy is a concept that cuts across many areas: for example, it can be concerned with something as simple as an opt-in or opt-out to an online marketing survey, or it can concern the tracking of one's online and offline activity. In the context of modern communication systems, privacy is usually concerned with one's personally identifying information, such as a subscriber's long-term credentials in the case of mobile telephony systems or an individual's health data in the case of Wireless Body Area Networks (WBANs) [70, 86].

1.1.2 Privacy in Mobile Telephony Systems

With the passage of time, we have come to rely more and more on our mobile phones. These devices have become part and parcel of our daily lives. It is now more important than ever before that end-user privacy be ensured in mobile telephony systems. The segment of mobile telephony systems where privacy controls are required the most is the radio access network, which is the most vulnerable to various privacy threats. In this thesis we analyze privacy controls for the 5G radio access network, point out the shortcomings and propose appropriate improvements.

1.1.3 Security and Privacy in WBANs

Another area of increasing concern is the security and privacy issues facing our health data. Health-related data has been shown [57] to be worth more than any other type of data record. This is because health information has a long shelf-life and is unalterable. With an ever-increasing aging population [111], remote healthcare seems to be the way forward. WBANs have emerged as a key technology for the realization of remote healthcare systems. IEEE Std 802.15.6 [3] is the international standard for WBANs.

Compliance requirements for health-related data, such as the U.S. based HIPPA Privacy Rule [117], try to ensure that privacy is upheld. But laws like HIPPA still need to be augmented with robust security standards concerning the underlying technologies such as WBANs, which should respect privacy as a right of the patient. Unfortunately, in the case of IEEE Std 802.15.6 this has not been the case. This thesis addresses this issue for IEEE Std 802.15.6.

1.1.4 Moving Towards Quantum-Secure Standards

Many examples of a preference for public-key cryptography can be witnessed in the latest communication standards. While it is true that key distribution and management are relatively easy to handle in the public-key domain, a reliance on public-key cryptography puts standards at risk of being vulnerable to quantum cryptanalysis. Moreover, key management becomes a "real" issue only when the number of users grows, which is not always the situation for some standards. For example, in IEEE Std 802.15.6, the maximum number of users that a WBAN is allowed to handle is 64 [3]. It is thus arguable, in this case, that the risk exposure greatly outweighs the purported benefits. Hence, it is worth considering whether it is possible to replace public-key cryptography with symmetric-key cryptography to avoid exposure to future quantum threats.

1.2 Research Contributions

The contributions of this thesis can be summarized as follows:

- This thesis presents a comprehensive literature review of subscription privacy on the 5G wireless interface. Moreover, various aspects of subscription privacy

are contextualized in chronological order, which provides an insight into the standards' development cycle. We highlight privacy issues that are yet to be addressed by the 5G standard. To address the identified privacy gaps, we also propose improvements for future versions of the 5G standard.

- The thesis also provides an analysis of the privacy attacks on 5G by [38]. The findings of our analysis contradict some of the claims made in [38]. Specifically, we show that the *activity monitoring attack* is infeasible to execute in 5G networks. We also demonstrate that the *location confidentiality attack* is a direct extension of an existing privacy vulnerability that exploits linkability of the authentication failure messages.

- We propose an alternative identification scheme which overcomes the limitations of the current public-key based identification mechanism of the 5G standard. We also develop a novel security framework titled Symmetric Updatable Private Authentication (SUPA) and provide a detailed formal security and privacy analysis of the proposed scheme in this framework.

- The existing identity protection mechanism of 5G and our alternative proposal are both vulnerable to downgrade attacks; i.e. an active attacker is able to force the connection down to one of the previous generations and exploit known vulnerabilities. This thesis also shows how a downgrade protection proposal for 5G [89] can be seamlessly integrated with our identity protection scheme to come up with a quantum-secure and downgrade-resistant identification mechanism for 5G.

- We propose two key agreement protocols for IEEE Std 802.15.6. These protocols, in addition to being efficient and provisioning advance security properties, also offer essential privacy attributes necessary for WBANs. The protocols are also quantum-secure as they are independent of any public-key based operations. We also develop a formal security and privacy model called Privacy Preserving Key Agreement (PPKA) in an appropriate complexity-theoretic framework and prove the proposed protocols secure in this model.

1.3 Thesis Structure

Chapter 2 provides the requisite technical background and preliminary material for Chapters 3 to 6. This chapter outlines the historical evolution of security and

privacy in various generations of mobile telephony standards. It also provides a description of the mobile telephony ecosystem and its pertinent security and privacy mechanisms.

Chapter 3 reviews the current status of subscription privacy on the 5G radio interface. Although 5G offers better privacy guarantees than its predecessors, this work highlights that there still remain significant gaps which need rectifying.

Chapter 4 analyzes two privacy attacks on 5G by Borgaonkar et al. [38]. We evaluate these attacks for their effectiveness, practicability and potency against 5G.

In Chapter 5, we present an alternative identity protection scheme for 5G which utilizes only symmetric cryptographic primitives and formally analyze its security and privacy properties.

In Chapter 6, we show how a downgrade protection proposal can be combined with our alternative identification scheme to provide a quantum-secure and downgrade-resistant private identification for 5G.

In Chapter 7, we further evolve the concepts and techniques which we develop in Chapter 5 to come up with two key agreement protocols for another international communication standard, the IEEE Std 802.15.6.

Finally, Chapter 8 provides concluding remarks and discusses future research directions.

1.4 Associated Publications

In the course of my PhD I have been fortunate to collaborate on my research questions with other talented researchers. In the following, I will expand on my role in each of these pieces of work, but wish to emphasize that all work was a collaborative effort, and could not have been possible without the contributions of all co-authors.

- Haibat Khan, Benjamin Dowling and Keith M. Martin. Identity Confidentiality in 5G Mobile Telephony Systems. In *4th International Conference on Security Standardization Research*, SSR 2018, Darmstadt, Germany, November 26-27, 2018. Proceedings Ed. by Cas Cremers and Anja Lehmann, LNCS Volume 11322, pp. 120-142, Springer, 2018 [83].

In this work, we presented our proposal for an alternative identity protection mechanism for 5G (Chapter 5). Being the main author, the overall conceptualization, research and writing of the draft was undertaken by myself.

- Haibat Khan, Benjamin Dowling and Keith M. Martin. Highly Efficient Privacy-Preserving Key Agreement for Wireless Body Area Networks. In *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*, TrustCom 2018, New York, NY, USA, August 1-3, 2018, pp. 1064-1069, IEEE, 2018 [82].

  This paper presented our key agreement protocols for WBAN standard IEEE Std 802.15.6 (Chapter 7). The major write-up, initial conceptualization and investigation was undertaken by me. Due to a limit on the number of pages of the conference proceedings, an abridged version of our work was published at this venue. A full version of this work is available online at https://eprint.iacr.org/2020/045 and is currently under submission.

- Haibat Khan and Keith M. Martin. On the Efficacy of New Privacy Attacks against 5G-AKA. In Proceedings of the *16th International Conference on Security and Cryptography*, SECRYPT 2019, Prague, Czech Republic, July 26-28, 2019. Proceedings Ed. by M. S. Obaidat and P. Samarati, pp. 431-438, SciTePress, 2019 [84].

  This paper detailed our analysis and evaluation of the privacy attacks against 5G (Chapter 4). Most of the work including research, investigation, drafting, etc. was undertook by myself.

- Haibat Khan and Keith M. Martin. A Survey of Subscription Privacy on the 5G Radio Interface - The Past, Present and Future. *Journal of Information Security and Applications*, vol. 53(102537), pp. 1-17, Elsevier, 2020 [85].

  This paper presents a state of the art survey of subscription privacy on the 5G wireless channel (Chapter 3). Majority of the work on this project was carried out by me.

Additionally, the following work was conducted during the course of this PhD but is not included in this thesis:

- Haibat Khan. An Identity based Routing Path Verification Scheme for Wireless Sensor Networks. *International Journal of Sensor Networks*, vol. 26(1), pp. 54-68, Inderscience Publishers, 2018 [81].

Chapter 2

Background and Preliminaries

This chapter provides the requisite technical background and introduces the notation and symbolism for the rest of the thesis. It starts by outlining the standardization process and the historical evolution of mobile telephony followed by a description of its ecosystem and pertinent security and privacy mechanisms. Thereafter, background regarding WBANs is described and the interconnection between mobile telephony and WBANs security and privacy problems is explained.

2.1 3rd Generation Partnership Project

The 3rd Generation Partnership Project (3GPP) is the de facto international body responsible for mobile telephony standardization. 3GPP unites seven telecommunications standard development organizations (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC), known as "Organizational Partners", and provides their members with a stable environment to produce the *reports* and *specifications* that define 3GPP technologies. The project covers cellular telecommunications technologies, including radio access, core network and service capabilities, which provide a complete system description for mobile telecommunications. The three Technical Specification Groups (TSG) in 3GPP are:

- Radio Access Networks (RAN);
- Services & Systems Aspects (SA);
- Core Network & Terminals (CT).

The TSGs are further divided into specialized Working Groups (WGs). Most of the work carried out on this thesis deals with the security and privacy aspects of 5G radio access technology and comes under the scope of SA WG3 (SA3), which

is a specialized group responsible for security and privacy of 3GPP systems. SA3 performs analysis of potential threats to 3GPP systems. Based on the threat analysis, SA3 determines the security and privacy requirements for 3GPP systems, and specifies the security architectures and protocols.

2.2 Evolution of Mobile Telephony Security

The First Generation (1G) of mobile telephony systems was based on analogue technology whose commercial deployment started during the 1980s. Various 1G technologies were deployed both regionally and globally, the most widespread of which was AMPS (Advanced Mobile Phone System). 1G systems offered no security to its users.

The first digital systems were introduced by the Second Generation (2G) of mobile telephony which additionally provided SMS and data services along with voice. 2G networks were commercially launched as Global System for Mobile Communications (GSM) in 1991 in Finland. As indicated in Table 2.1, 2G plugged glaring holes in 1G security, offering payload encryption and authentication of mobile subscribers to the networks.

The Third Generation (3G) of mobile telephony introduced in 2001 as Universal Mobile Telecommunications Service (UMTS) upgraded 2G systems for faster data transfer speeds (at least 144 kbit/s) and paved the way for mobile broadband access. The security improvements in 3G systems were also significant. One-way authentication was transformed into mutual authentication between mobile subscribers and their service providers. Moreover, publicly-known encryption and integrity algorithms with improved key lengths (128-bits) were introduced.

The Fourth Generation (4G) systems, which commercially debuted as Long Term Evolution (LTE) in 2009 in Norway, re-utilized a few of the 3G encryption and integrity algorithms. One security improvement that 4G offered was that the authentication protocol additionally assured the mobile subscriber of the identity[1] of its service provider. Moreover, from 4G onwards, use of different encryption and integrity algorithms resulted in the derivation of distinct keys, unlike earlier generations.

The latest Fifth Generation (5G) of mobile telephony systems has recently been

[1] In 3G, the user was only assured of the legitimacy of the service provider.

Table 2.1: Security evolution in mobile telephony generations.

Security Aspect	1G	2G	3G	4G	5G
Authentication	No	One-way	Mutual	Proves the exact network to the user	Prevents roaming billing fraud
Cipher Key Length	No	54/64-bits	128-bits	128-bits	128-bits
Encryption Algorithm Strength	N/A	Weak	Strong	Strong	Strong
Public Algorithms	N/A	Not Public	Public	Public	Public
Distinct Keys for Algorithms	No	No	No	Yes	Yes
Signalling Integrity	No	No	Yes	Yes	Yes
Traffic Integrity	No	No	No	No	Yes

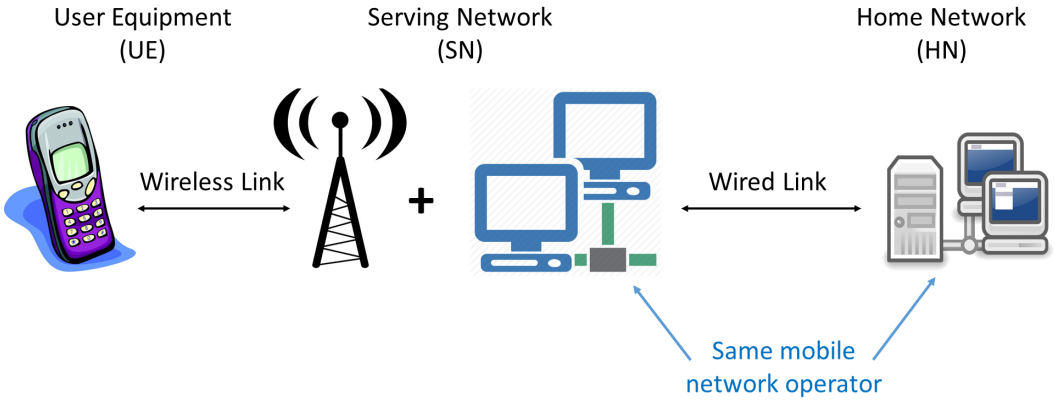
standardized and is now undergoing commercial deployment. The significant security enhancements offered by 5G systems are the ability to integrity protect user plane traffic (payload) in addition to the control plane traffic (signalling), and the protection from roaming billing fraud by giving increased control to the home network during the execution of the authentication and key agreement protocol. In 5G, the home network of the mobile subscriber is also provided with a proof of authentication after a successful user authentication (see §2.7 for more details).

2.3 Mobile Telephony System Architecture

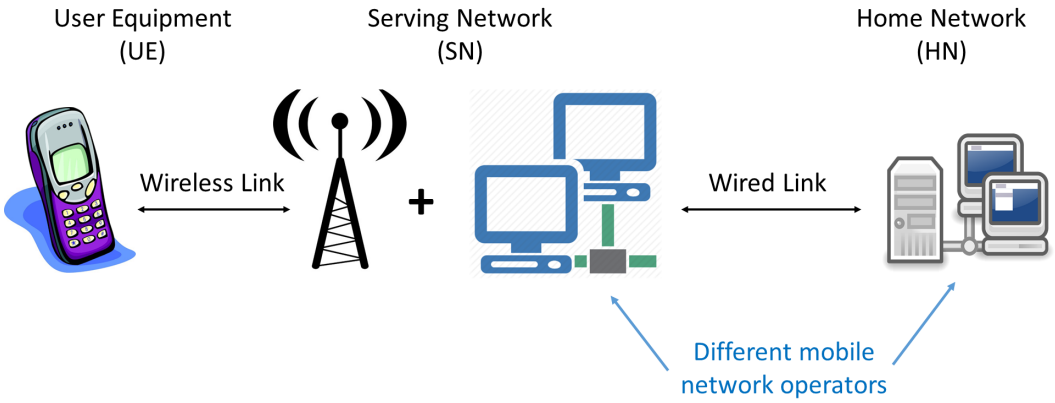
The mobile telephony architecture consists of three main domains; Home Network (HN), Serving Network (SN) and User Equipment (UE) (see Figure 2.1). The *subscribers* carry UE, which typically refers to Mobile Equipment (ME) (the phone) containing a Universal Integrated Circuit Card (UICC) (the SIM card). The HN domain represents the network functions that are conducted at a permanent location regardless of the location of the subscriber. The HN is where a subscription initially gets registered. It stores the subscribers' credentials and is responsible for management of subscription information. The SN domain is the part which provides the subscribers access to the telephony network and its services. It represents the network functions that are local to the user's access point and thus their location changes when the user moves. The SN is responsible for routing calls and transport of user data/information from source to destination. It has the ability to interact

with the HN to cater for user-specific data/services.

Often UEs may have to operate in areas where their operators have no network coverage (i.e., base stations). In such scenarios called *roaming*, other service providers, who have a roaming agreement with the subscriber's operator, provide SN services. Hence, in this paper, we treat SN as a semi-trusted entity to whom a subscriber's long-term credentials can not be exposed (barring a few exceptions). Note that according to the 3GPP standard [14], HNs and SNs are further divided into logical sub-entities. The security and privacy properties being discussed in this thesis do not require this level of granularity.



(a) When not roaming, both HN and SN belong to the same mobile network operator.



(b) When roaming, the SN and HN belong to distinct mobile network operators.

Figure 2.1: The mobile network architecture. The channel between UE and SN is initially unprotected while that between SN and HN is assumed to be protected.

It is within the UICC that the application Universal Subscriber Identity Module (USIM) runs. The USIM represents the relationship between a subscriber and its issuing HN. During a subscription registration, the HN stores the subscriber's long-

term identifier, Mobile Station International Subscriber Directory Number (MSISDN) (the telephone number) and other subscriber related data, including a 128-bit secret key K and 48-bit monotonically increasing counters called Sequence Numbers (SQNs), within the USIM. The key K never leaves the USIM and any processing that requires key K as an input is executed within the USIM. These SQNs are utilized for the purpose of replay prevention. While an SQN should be synchronized between the UE and HN, sometimes it may become out-of-sync due to the loss of messages on the wireless channel. We therefore use SQN_{UE} and SQN_{HN} to refer to the state of SQN in UE and HN respectively. These subscription parameters are also stored within the HN's database and form the basis of a security context between UEs and HNs and by extension (during roaming) between UEs and SNs. The SNs provision services to UEs after establishment of a secure channel between them with help of the HNs.

2.4 Identifier Types and Terminologies

In mobile telephony systems, networks allocate to each subscriber a unique long-term identifier, known up to 4G as a International Mobile Subscriber Identity (IMSI) and since 5G as a Subscription Permanent Identifier (SUPI). A SUPI, as defined in 3GPP TS 23.501 [15], is usually a string of 15 decimal digits and acts as the long-term identifier of an individual subscriber. The first three digits represent the Mobile Country Code (MCC), while the next two or three form the Mobile Network Code (MNC), which identifies the network operator. The length of the MNC field is a national affair. The remaining (nine or ten) digits are known as the Mobile Subscriber Identification Number (MSIN) and represent the individual user of that particular operator. Each decimal digit of the SUPI is represented in binary by using the Telephony Binary Coded Decimal (TBCD) encoding [12].

Authentication between a user and its service provider is based on a shared symmetric key (details in §2.7), which means it can only take place after an initial user identification. However, if the IMSI/SUPI values are sent in plaintext over the radio link for this purpose, then subscribers can be identified, located and tracked using these permanent identifiers. To avoid this privacy breach, subscribers are assigned temporary identifiers called Globally Unique Temporary User Equipment Identities (GUTIs) by the SNs. A GUTI uniquely and globally identifies a particular subscriber. These frequently-changing temporary identifiers are then used for

identification purposes over the wireless link before the establishment of a secure channel.

The International Mobile Equipment Identity (IMEI), which uniquely identifies the ME, is a string of 15 digits. If the IMEI is sent in plaintext over the radio interface then it could compromise user privacy as it is also uniquely identifying from a subscription viewpoint. However, the 3GPP specifications prohibit a UE from transmitting the IMEI until after establishment of a secure channel with the network [118].

2.5 Security Assumptions

2.5.1 Assumptions on Channels

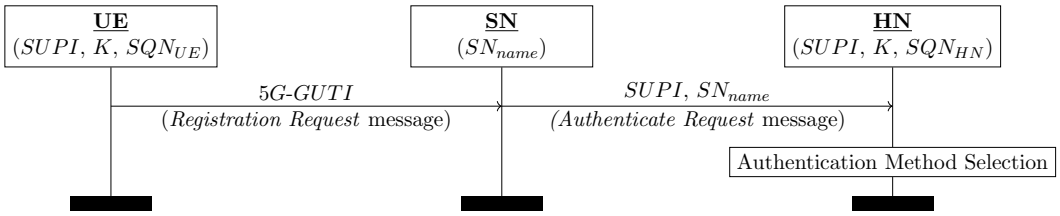
According to 3GPP TS 33.501 (sub-clause 5.9.3) [14], the channel between SN and HN should provide confidentiality, integrity, authentication and replay prevention. The channel between UE and SN, essentially being a wireless one, is subject to eavesdropping, interception and injection of messages by malicious third parties.

2.5.2 Assumptions on Parties

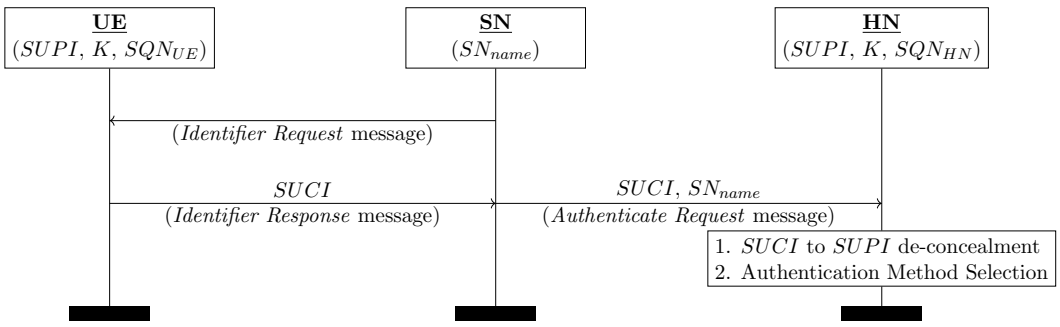
The USIM and its associated HN are fully trusted entities. The shared secret data being stored by these two entities is assumed to be protected from third parties. Specifically, the UICC (upon which USIM is stored) is considered to be a tamper-resistant security module whose contents cannot be read by a malicious entity. MEs are semi-trusted devices because the long-term key K of the USIM is never revealed to them. SNs are also semi-trusted entities in the sense that during the secure channel establishment the long-term shared secret key K and sequence numbers SQN should not be revealed to them while SUPI is provisioned to them. The provisioning of SUPI is essential for accurate billing purposes.

2.5.3 Assumptions on Cryptographic Functions

All the cryptographic functions (detailed in §2.7) are assumed to provision both confidentiality and integrity protection to their respective inputs.



(a) Authentication procedure utilizing 5G-GUTI.



(b) Authentication procedure utilizing SUCI.

Figure 2.2: Initiation of authentication procedure.

2.6 Initialization of Authentication

As we will see in §2.7, secure channel establishment between subscribers and their service providers is done via challenge-response protocols which are based upon the shared secret key K. Thus, before such protocols can be executed, it is imperative that the service provider correctly identifies the subscriber with whom this channel needs to be established. 3GPP TS 33.501 (sub-clause 6.1.2) [14] details the procedures for this subscription identification and selection of the subsequent authentication method. The same is being depicted pictorially in Figure 2.2.

The SN may initiate an authentication with the UE during any procedure establishing a connection with the UE. The UE sends the SN either the 5G-GUTI in a *registration request* message (Figure 2.2a) or the Subscription Concealed Identifier (SUCI) as a response to an *identifier request* message (Figure 2.2b). SUCI is a randomized public-key encryption of SUPI (see §5.4 for details). In the case of a 5G-GUTI, the SN extracts the corresponding SUPI from its database and forwards it along with its global identity Serving Network Name (SN_{name}) to the HN in an *authenticate request* message. Otherwise, the SUCI is sent instead of the SUPI. Upon receipt of the *authenticate request* message, the HN checks whether the SN is entitled to use the serving network name in the request message by comparing the

Table 2.2: Description of 5G-AKA parameters.

Parameter	Content/Description
$RAND$	Random Challenge
AK	Anonymity Key
CK	Confidentiality Key
IK	Integrity Key
RES	Response
MAC	Message Authentication Code
$CONC$	Concealed Sequence Number
$AUTN$	Authentication Token
$AUTS$	Resynchronization Token
$XRES$	Expected Response
$HRES/HXRES$	Hash of $RES/XRES$
K_{AUSF}	Intermediate Key
K_{SEAF}	Anchor Key

incoming serving network name with the expected serving network name. The HN stores the received serving network name temporarily. If the SN is not authorized to use the serving network name, the HN responds with a "serving network not authorized" message. If the SUCI is received in an *authenticate request* message by HN, it de-conceals the SUPI from it and chooses the authentication method based upon its policy.

2.7 The 5G-AKA

The security of communication between telephony subscribers and their service providers requires mutual authentication and key agreement. In 5G systems, these requirements are fulfilled by either EAP-AKA' or 5G-AKA, which are both Authenticated Key Agreement (AKA) protocols. EAP-AKA' and 5G-AKA are quite similar, with identical message flows, but with a little difference in way the various keys are derived. We therefore only consider 5G-AKA in this thesis. 3GPP TS 33.501 (sub-clause 6.1.3.2) [14] defines the details of the 5G-AKA protocol. The security of 5G-AKA is based upon the shared symmetric key K, while SQN provisions replay protection. To initiate authentication, the UE sends the SN either the 5G-GUTI in a *registration request* message, or the SUCI as response to an *identifier request* message as explained in §2.6.

Figure 2.3 shows the 5G-AKA and its associated failure mechanisms. Table 2.2 details the various acronyms used in Figure 2.3. In this figure, $RAND$ is a uniformly

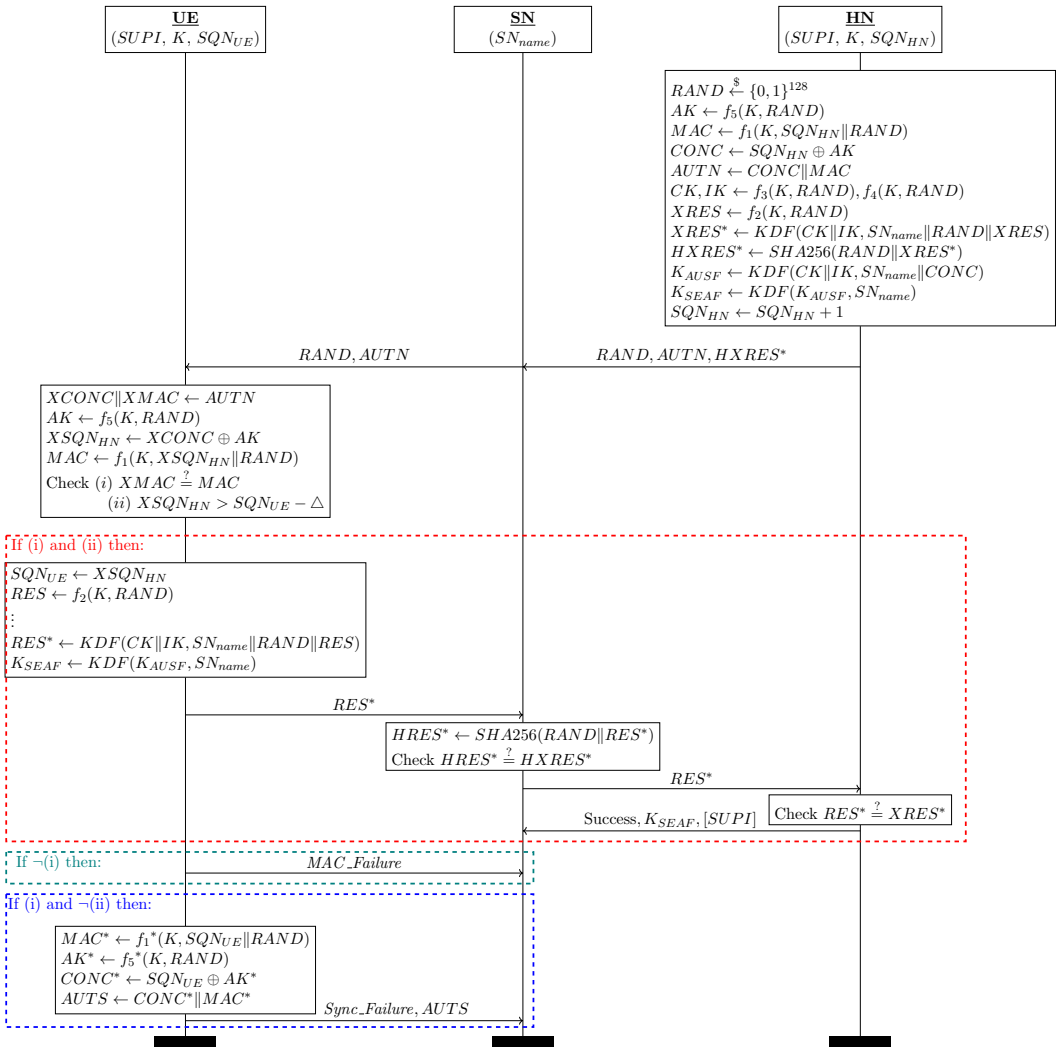


Figure 2.3: The 5G-AKA protocol and its associated failure mechanisms.

chosen 128-bit random number and functions f_1,..., f_5, $f_1{}^*$ and $f_5{}^*$ are symmetric key algorithms. Functions f_1, f_2 and $f_1{}^*$ act as message authentication algorithms, while f_3, f_4, f_5 and $f_5{}^*$ are used as key derivation algorithms. Key derivation is performed using the Key Derivation Function (KDF) specified in 3GPP TS 33.220 [11]. A successful 5G-AKA culminates in the derivation of the anchor key K_{SEAF} by both SN and UE from which further keys for subsequent communication are derived. The two cases of authentication failure for the 5G-AKA are as follows:

1. **MAC_Failure:** As the first step in authentication confirmation, the UE checks whether the received MAC value is correct or not. In case of a failure (Case ¬(i) in Figure 2.3), the UE replies with a $MAC\_Failure$ message back to the SN.

2. **Sync_Failure:** After MAC verification, the UE checks the freshness of the sequence number SQN_{UE} received in the authentication challenge. In case of this failure (Case (i) and ¬(ii) Figure 2.3), it responds with a *Sync_Failure* message along with a re-sync token $AUTS$. Note that in Figure 2.3, the sequence number freshness check is denoted by $XSQN_{HN} > SQN_{UE} - \triangle$. What this actually means is that there is some "window" of size $\triangle$ within which sequence numbers smaller than the current sequence number of UE will be accepted given they previously had not been received by the UE. This mechanism is there to handle out-of-order delivery of challenge messages from HN to UE.

During the execution of 5G-AKA, it is crucial that SQN is protected from an eavesdropper during the exchange of messages between the UE and SN, as its exposure may lead to the compromise of the identity and location of a subscriber. We will see in §3.5.2 how SQN leakage can manifest into privacy vulnerabilities. Also note from Figure 2.3 that at the culmination of a successful 5G-AKA, the HN provides the SUPI of the UE to the SN. This is required essentially for two main purposes - accurate billing and Lawful Interception (see §2.9). The SUPI is later also used as an input to the key derivation functions between UE and SN. This ensures that the SUPI value provisioned by the HN is the one claimed by the UE, otherwise the communication breaks down.

2.8 Paging Messages

When a UE does not have any ongoing data transmissions, it enters an *idle* state in order to preserve energy. If delivery of a network service like a call or SMS needs to be delivered to the UE, the network probes the *idle* UE by sending a "paging" message and the UE responds correspondingly. The paging procedure works because even when in the *idle* state, the UE keeps on monitoring for the paging message at certain device-specific time intervals. The device is able to preserve battery because, at other times, it switches off its receiver. The idle UE decodes these broadcast probes and if it detects its identity in these messages, it randomly acquires an available radio channel and requests the concerned base station for "connection setup" for exchange of further signalling messages. It is worth mentioning that a new authentication may be initiated by the SN after the UE responds to a paging message.

2.9 Lawful Interception

Lawful interception (LI) refers to the facilities in telecommunication networks that allow law enforcement agencies with court orders or other legal authorization to selectively wiretap individual subscribers. Usually network data collection under LI is for the purpose of analysis or evidence. Such data generally consists of signalling or network management information or, in fewer instances, the content of the communications. The collection of data could or could not be in real-time and can be performed in either core or edge network.

3GPP TS 33.126 [25] specifies various LI requirements for telecommunication operators. As such not all requirements in this document will apply in all national jurisdictions or to all 3GPP operator deployments. Various LI architectures and functions are detailed in 3GPP TS 33.127 [24]. This document provides an LI architecture supporting both network layer based and service layer based interception. 3GPP TS 33.128 [26] describes the protocols and procedures required to perform LI within a 3GPP network. It addresses both internal interfaces used internally with a 3GPP network and external handover interfaces used to handover intercepted communications to law enforcement. It describes the detailed targeting of communications in each point of interception within a 3GPP network and the information that a point of interception needs to be able to capture. Furthermore, the detailed data formats for both the internal and external interfaces are also defined.

2.10 Wireless Body Area Networks

Wireless Body Area Networks (WBANs) consist of miniaturized computing devices with the aim to provide low power, short range, and extremely reliable wireless communication within the surrounding area of the human body, supporting a vast range of data rates for different applications [44]. These devices talk to a designated centralized node (Hub) which further communicates with external networks via a Gateway [110]. The general layout of a typical WBAN is illustrated in Figure 2.4. Note that the Hub and Gateway are functionally two separate entities, but are usually combined into a single physical node.

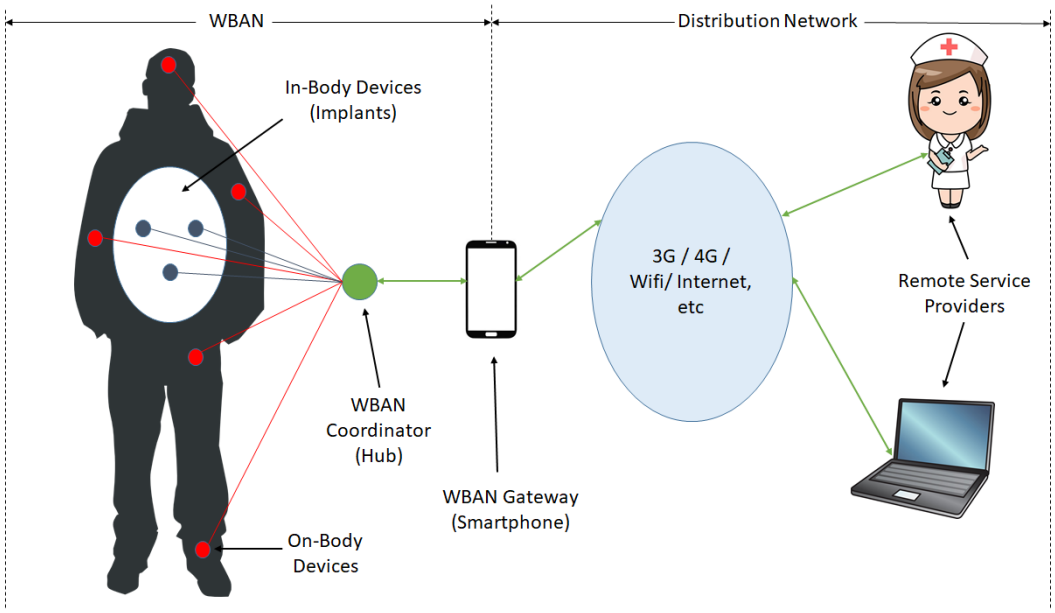


Figure 2.4: Generic architecture of a typical WBAN.

2.11 WBAN System Model

We now describe a system model suitable for the deployment scenarios of WBANs. In this model, a System Administrator (SA) initializes the network. The network is composed of three types of nodes: a Hub Node (HN[2]), Intermediary Nodes (IN) and Normal Nodes (N). As the HN is usually a resourceful device with better hardware protection mechanisms in place, we assume it to be trusted and its long term secret *master key* (k_{HN}) to be protected. As the role of HN is usually undertaken by a modern smartphone in a generic WBAN, this argument is supported aptly by the real-world example of the "FBI-Apple encryption dispute" [2] where, even for resourceful parties like government agencies, it is not easy to crack into a smartphone. Normal nodes N are resource-constrained and their transmission range is assumed to be limited; in particular, they are not always able to communicate directly with HN. Intermediary nodes IN are also located in and around the body but, at a particular time instance, are in direct communication with both N and HN, thus acting as intermediary nodes for the purpose of relaying traffic between HN and N when required. We assume a Dolev-Yao [56] adversary $\mathcal{A}$ who can listen, modify and synthesize any messages of his choice in this model.

[2]Note that the notation HN has also been used in this thesis to refer to the *home network* of mobile telephony systems. However, within the context of WBANs, HN refers to the *hub node*.

2.12 IEEE Std 802.15.6

Mindful of the peculiarities of communicating in and around the human body, the IEEE published the standard IEEE Std 802.15.6 [3] for WBAN communications in 2012. Being part of the 802 series of IEEE standards, which comprises of a family of networking standards that cover the physical layer specifications of technologies from Ethernet to wireless, IEEE Std 802.15.6 describes the physical (PHY) layer and the medium access control (MAC) sublayer for WBAN communication in accordance with the IEEE 802 reference model. The standard contains three specifications of PHY layer; narrowband, ultra wideband and human body communications. Direct communication between a node N and a hub node HN transpire at the PHY layer and MAC sublayer as specified in this standard. Additionally, the standard also describes the frame formats and various functional elements associated with the MAC sublayer. As high-power transmissions are harmful to humans and WBAN nodes are energy constrained, this standard provisions an optional two-hop communication architecture via a relay node to enable nodes to communicate with the Hub node.

Message security services occur at the MAC sublayer, and security key generation functionality can take place inside and/or outside the MAC sublayer. In addition to conventional security guarantees, privacy is of utmost importance for typical target application areas of WBANs such as healthcare and the military [136]. The elliptic-curve based key agreement (termed as *security association* within the standard) protocols of IEEE Std 802.15.6 have been shown to have security weaknesses [134], but also do not provide the privacy features that should be expected of a WBAN [102]. In Chapter 7 we propose two key agreement protocols for IEEE Std 802.15.6 which in addition to the requisite security properties, also offer privacy guarantees. Next, we discuss the interconnection between the works carried out in Chapters 5 and 7.

2.13 Two Sides of the Same Coin

As one progresses through this thesis, they may notice the similarities between the techniques utilized for coming up with a private user-identification scheme for 5G (Chapter 5) and the privacy-preserving key agreement protocols for IEEE Std 802.15.6 (Chapter 7). Both utilize only symmetric cryptographic primitives in a much similar style and manner to cope with slightly varying problems. The reason for this is that the threat model, associated trust assumptions, security and privacy objectives, etc.

are quite similar for both scenarios, which enabled us to import techniques from one setting to handle the problems of the other. In both settings, there is a central trusted entity (*home network* for the mobile telephony and *hub node* for WBAN) which can be bootstrapped securely, both demand similar kind of privacy guarantees like *anonymity* and *unlinkability* and both want to avoid the use of public-key cryptography. In the case of mobile telephony, we are only concerned with the user-identification in a private manner which forms the basis of the later key agreement via a separate 5G-AKA. However, in the case of WBANs, we go a step forward and transform the private identification scheme into a full fledged key agreement protocol.

Next we explain the evolution and developmental connections between the results of Chapters 5 and 7. From a temporal viewpoint, the work in Chapter 7 was carried out before than that of Chapter 5. The initial inspiration for the work of Chapter 7 came from Li et al.'s scheme [103]. After analyzing and later improving Li et al's scheme, it became clear that such techniques could also be utilized for the private identification problem in mobile telephony networks. Note that by that time, the public-key based identity protection scheme (see Section 5.4.1) was yet not published by 3GPP. Though, the basic premise was same as that of WBAN, in mobile telephony the various cryptographic guarantees were required to be ensured via the already available primitives.

2.14 Chapter Summary

This chapter provided the background information for the mobile telephony systems. It presented an overview of evolution of security through different generations of mobile telephony. Further, the security architecture was outlined and pertinent security and privacy mechanisms were reviewed. Specifically, various types of user identifiers utilized within the mobile telephony systems along with the associated user identification and authentication protocols were detailed. Thereafter, an introduction to WBANs and its international standard IEEE Std 802.15.6 was provided

Chapter 3

State of Subscription Privacy in 5G

This chapter reviews the current state of subscription privacy in 5G systems. The scope of the privacy study undertaken is limited to the wireless part of the 5G system which occurs between the service provider's base station and the subscriber's mobile phone.

3.1 Introduction

Mobile telephony subscribers' personal information has become an attractive target for online advertisements and other connected industries. Besides the commercial arena, the Edward Snowden revelations show that national intelligence agencies also collect telephony subscribers' personal information on an unprecedented scale [67]. Apart from the danger that this personal information is utilized for nefarious political agendas, it may also be misused for personal advantages. Thus, privacy has turned out to be a primary consideration for end users when selecting and using a telephony service today. From a regulatory compliance perspective, the EU General Data Protection Regulation (GDPR) [137] obligations for protecting personal data of subscribers are directly applicable to mobile telephony operators. With penalties that can reach as high as EUR 20 million or 4 percent of total worldwide annual turnover, there is a huge financial risk for mobile operators in the event of potential non-compliance. Hence, protecting end-user privacy is all the more important for the latest international mobile telephony standards such as 5G.

3GPP released the first documents pertaining to 5G at the end of the year 2017. The development of the 5G system was planned in two phases: 5G Phase 1 (formally called Release 15) and 5G Phase 2 (formally Release 16).As 5G Release 15 – the first full set of 5G standards – was frozen [1] in June 2019 (see Figure 3.1), this seems to be an appropriate occasion to undertake a comprehensive review of one of the most

[1] After "freezing", no additional functionality can be added to a Release.

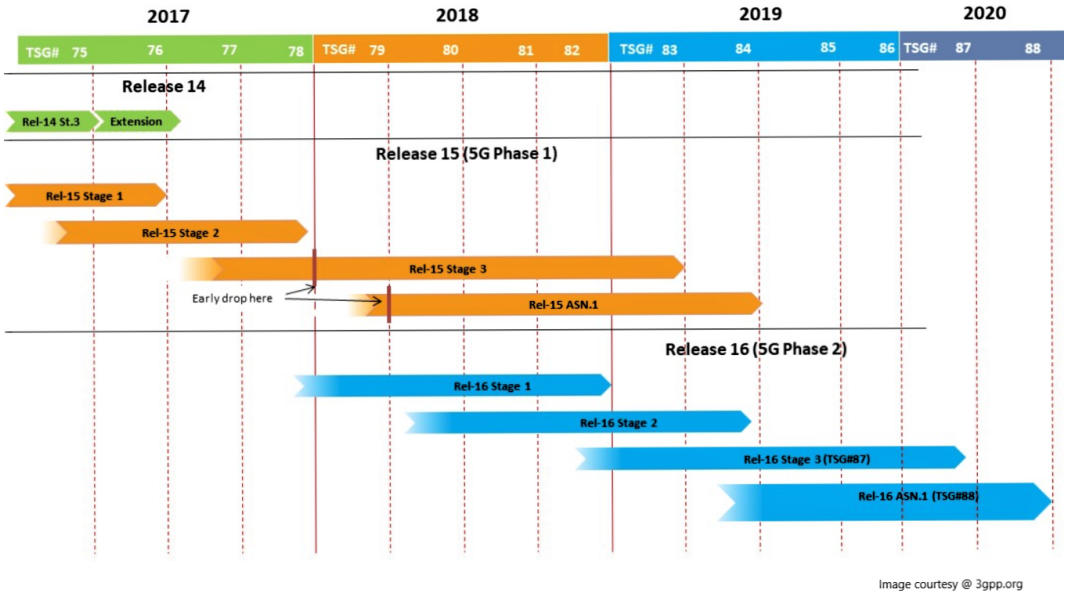


Figure 3.1: 3GPP time-lines pertaining to various Releases.

prominent privacy aspects of 5G based mobile telephony, i.e., subscription privacy on the wireless channel.

5G security and privacy documentation [14] often refers to previous generations for elaboration of various security and privacy requirements. The same is true in the case of subscription privacy where Release 15 refers to 3GPP TS 33.102 [9] for the requirements which are listed below:

- **User Identity Privacy:** The permanent identity of a user to whom a service is delivered cannot be eavesdropped on the radio access link.
- **User Location Privacy:** The presence or the arrival of a user in a certain area cannot be determined by eavesdropping on the radio access link.
- **User Untraceability:** An intruder cannot deduce whether different services are delivered to the same user by eavesdropping on the radio access link.

An important point to note here is that the use of the phrase "cannot be eavesdropped" in the above statements should not be misinterpreted if it only refers to a passive adversary 'eavesdropping' on the radio interface. This certainly is not the case here and a few previously published papers [33] fell prey to this misnomer. 3GPP has always considered active adversaries for its security and privacy scenarios. A pertinent example of this is the 3GPP study TR 33.899 [8] which was conducted to

collect, analyze and further investigate potential security threats and requirements for 5G systems and contains explicit references to active adversaries.

In this chapter, we provide an overview of the state of subscription privacy on the 5G radio interface. Keeping the aforementioned privacy objectives in mind, this chapter evaluates, systematizes, and contextualizes the requisite aspects of 5G subscription privacy in three chronological categories; past, present and the future. The *past* category looks at the state of subscription privacy before the advent of 5G Release 15. In *present*, the improvements provisioned to user privacy by Release 15 are explored. Finally, the *future* category discusses the privacy aspects which still could be improved in subsequent Releases.

The rest of the chapter is organized as follows: §3.2 discusses the scope of the privacy study while §3.3 details the *past* vulnerabilities. 5G Release 15 improvements are discussed in §3.4. §3.5 highlights the outstanding issues and new vulnerabilities. §3.6 discusses the related work and §3.7 concludes the chapter.

3.2 Scope of the Study

There are three aspects which play a pivotal role in defining the scope of the study undertaken in this chapter:

- We confine the privacy study undertaken in this chapter to the wireless part of the 5G system. This is primarily because this medium is open and can easily be exploited by any malicious party and, as a result, is the most vulnerable.
- In this study only those aspects of subscription privacy are discussed which come under the purview of 3GPP. Modern-day smart phones have evolved into powerful devices with functionality that goes beyond just telecommunications. These multitasking devices are now being utilized for all sorts of computational purposes which may or may not affect the end-user privacy that 3GPP is trying to protect. There are numerous other sources of leakage affecting user privacy such as Wi-Fi [74], Bluetooth [75], etc. which do not fall under the purview of 3GPP. We do not consider privacy leakages via these other sources in this work.
- Lastly, as work on 3GPP Release 16 (Phase 2 of 5G) is still under active development, we do not consider the ever-evolving Release 16.

Table 3.1: Summary of privacy attacks in the previous generations.

Section	Attack	Type			Attacker Capabilities					Generation		
		Identity Disclosure	Location Leak	User Traceability	Radio Passive	Radio Active	IMSI	MSISDN	TMSI / GUTI	2G	3G	4G
3.3.1	IMSI-catching [118], [120], [108], [53], [52], [113], [104], [63]	●	●	●	●	●	○	○	○	●	●	●
3.3.2	(Raw) IMSI-probing [51]	○	●	●	●	◐	○	●	○	●	●	●
3.3.3	Unauthenticated IMEI Request [120], [53], [113]	●	●	●	●	●	○	○	○	●	●	◐
3.3.4	GUTI Persistence [32], [129]	○	●	●	●	○	◐	◐	○	●	●	●
3.3.5	GUTI-MSISDN Mapping [129], [97], [114], [72]	○	●	●	○	◐	○	●	○	●	●	●
3.3.6	C-RNTI based Tracking [77]	○	●	◐	●	◐	○	●	○	●	●	●
3.3.7	GUTI Reallocation Replay Attack [32], [30]	○	○	●	●	●	○	○	○	●	●	●
3.3.8	Localization through Measurement Reports [129], [61]	○	●	●	●	●	?	●	●	○	○	●
3.3.9	IMSI-paging Attack [31], [129], [32], [132]	○	●	●	●	●	◐	◐	○	●	●	●
3.3.10	ToRPEDO Attack [73]	◐	●	●	●	◐	○	●	○	●	●	●
3.3.11	AKA Protocol Linkability Attack (LFM) [32], [31], [39]	○	○	●	●	●	○	○	○	○	●	●

Legend: ● = yes, applicable ◐ = partially/limited/optional ○ = no, not applicable ? = property unknown

3.3 The Past - Inherited Challenges

The first and foremost task for 5G Release 15 was to address the privacy vulnerabilities that existed in the previous generations. Hence, before we discuss the improvements offered by Release 15, we take a look at the vulnerabilities that already existed in the early generations that affect subscription privacy on the radio channel. Table 3.1 provides a summary of the attacks on subscription privacy in earlier generations.

3.3.1 IMSI-catching

As mentioned in §2.4, for obvious privacy reasons, GUTI is utilized for subscription identification purposes over the wireless interface before the establishment of a secure channel. However, there are certain situations where authentication through the use of these temporary identifiers is not possible. For instance, when a user registers with a network for the first time and is not yet assigned a temporary identifier. Another case is when the network is unable to resolve the IMSI from the presented GUTI. An active man-in-the-middle adversary can intentionally simulate this scenario to force an unsuspecting user to reveal its long-term identity. These attacks are known as "IMSI-catching" attacks [104] and persist in mobile networks including LTE [108, 118].

IMSI-catching attacks have threatened all generations of mobile telephony for decades [63]. In IMSI-catching, the attacker through the use of *identifier request* messages (§2.6) can get the identities of every subscriber present in the attack area. The attacker needs no previous assumption of who might be there, and needs no previous information about the victim. Thus, it is a powerful attack, breaching the subscription privacy completely. IMSI-catching is well documented as a *Key Issue*[2] in 3GPP TR 33.899 (sub-clause 5.7.3.2) [8].

3.3.2 (Raw) IMSI-probing

In its discussions, 3GPP distinguishes between "IMSI-catching" and "IMSI-probing". IMSI-probing is where an attacker already knows the subscription identity, like an IMSI or an MSISDN plus some associated information, and wants to find out whether the subscriber with this identity is present in a given area. This is a far less powerful attack than IMSI-catching. There are many possible ways to carry out such an attack, for example by sending a bunch of (if possible silent [51]) SMSs or other "activity triggers" to the target MSISDN and seeing whether there is a corresponding flurry of signalling in the cell being tested. Preventing all sorts of IMSI-probing attacks would be difficult and would involve a lot of overhead, such as including extensive dummy signalling to conceal when the real signalling happens. Consequently, it was not thought worthwhile to try addressing by 3GPP.

3.3.3 Unauthenticated IMEI Request

In GSM and UMTS systems, it was possible for an attacker to request the subscriber for its IMEI via an unauthenticated *identity request* message [53, 113, 120]. However, from LTE onwards, such provisions were removed and now the network can only request the user for its IMEI after establishment of a secure channel between them [16]. However, under certain special circumstance, e.g., when the UE has no IMSI or no valid GUTI during *emergency attach*, the IMEI is sent before a security context is activated. This is to restrain the misuse of ME for placing invalid emergency calls [6].

[2] *Key Issue* is the terminology used in 3GPP studies for potential security or privacy problem.

3.3.4 GUTI Persistence

Temporary subscriber identifiers like GUTI are used as a privacy measure to mitigate subscription identification and tracking by eavesdroppers on the radio link, making it harder to track the location or activity of a particular subscriber. In LTE, the updating of GUTI is recommended on the following occasions:

- When the SN gets changed or during a new *attach* procedure;
- During a Tracking Area (TA) update;
- When the SN issues a "GUTI reallocation command".

The major problem with the mechanism of GUTI allocation in the current LTE system is that it is up to the SN policy configuration when, and if at all, to reallocate the GUTI. It is also possible for the SN to keep (re)allocating the same GUTI to the UE. The UE neither takes part in the generation of the GUTI nor verifies the freshness of the newly allocated GUTI. This opens up possibilities for either poor implementations or poor configuration that keeps the GUTI the same for a long time. Evidence of these poor practices has been found in real mobile network operators [129, 32] where the operators tend not to frequently update the GUTI on these occasions. The reason ascribed to such practices is to avoid the signalling storms [27] within the networks. In LTE networks, acquiring or tracking the temporary subscription identifiers has been one of the most prominent attack strategies in compromising the subscription privacy [129]. GUTI persistence has been identified as a *Key Issue* in 3GPP TR 33.899 (sub-clause 5.7.3.1) [8].

3.3.5 Mapping between GUTI and MSISDN

These attacks are somewhat related to the IMSI-probing ones, but are more fine-grained. In these attacks, the attacker starts with similar assumptions about knowing one of the subscription long-term identities and the aim is to locate and then further trace that subscriber. The attack uses the usual techniques of either initiating phone calls [97] or sending silent SMSs [114] to the target MSISDN. This results in triggering of their paging procedures which ultimately leads to a mapping between the known identity (usually MSISDN) and the GUTI [72]. This enables an attacker to track a particular subscriber over a long duration due to infrequent updating of GUTI in LTE (as already detailed in §3.3.4). Note that in these attacks

paging messages are sought by the attackers instead of looking out for a generic signalling flurry.

3.3.6 C-RNTI based Tracking

The Cell Random Network Temporary Identifier (C-RNTI) is a physical layer 16-bit identifier unique within a given cell and is assigned to each device during the "Random Access Procedure" (see Section 3.3.9 for details). Passive analysis of real LTE traffic has revealed that the C-RNTI is included in the header (in unencrypted form) of every single packet [77]. This leads to linking of the radio traffic (both user and control plane) by a passive adversary. Further mapping to a user's GUTI or MSISDN is trivial and can be undertaken via the use of silent text messages. Through tracking of the C-RNTI value, an attacker can easily determine how long a given user stays at a given location.

Further analysis of captured LTE traffic has revealed that during mobility handover events these physical layer identifiers can be linked together. This leads to traceability of users when they move from cell to cell. This was because the captured handover triggering messages were sent in the clear. According to the response of the standardization bodies, these messages are not suppose to be in the clear.

3.3.7 GUTI Reallocation Replay Attack

As explained in §2.4, subscribers communicate with the networks using GUTIs as their identifiers for privacy purposes. To avoid traceability of subscribers based upon GUTI, it is imperative that these temporary identifiers are updated frequently. To update the GUTI, the mobile networks use a process called "GUTI Reallocation Procedure" (sub-clause 5.4.1 of TS 24.301 [22]). Figure 3.2 depicts this procedure as defined for LTE in [22]. In this figure, $oGUTI$ depicts the old GUTI and $nGUTI$ is the new GUTI, while CK is the "confidentiality key". The procedure works as follows:

- The UE identifies itself to the network on a dedicated channel via its currently allocated temporary identifier $oGUTI$.
- The network identifies the UE and establishes the means of ciphering for subsequent communication.

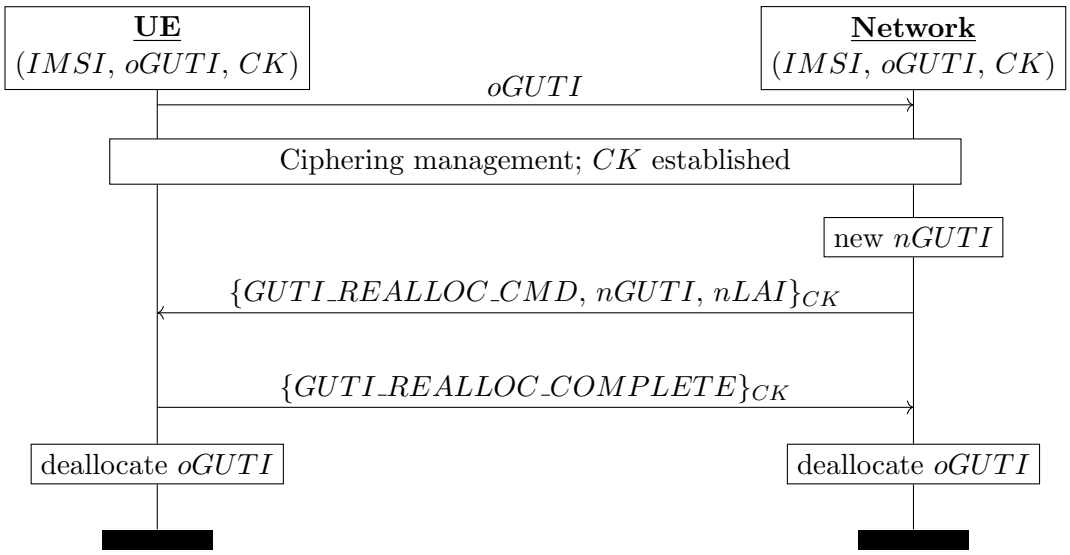


Figure 3.2: GUTI reallocation procedure.

- Thereafter, a new GUTI ($nGUTI$) is sent to the UE in a message encrypted with CK via a $GUTI\_Reallocation\_Command$. If required, this message may also contain the identity of the current location area ($nLAI$).
- Upon receipt of the GUTI reallocation command, the GUTI replies via the $GUTI\_Reallocation\_Complete$ message to acknowledge receipt of the new GUTI.

If the network does not receive the expected acknowledgment from the UE, it maintains both $oGUTI$ and $nGUTI$ for the concerned IMSI. The standard defines two methods for the means of ciphering; i.e. for the establishment of the confidentiality key CK: (1) either a new key is established via the authentication procedure or (2) a previously established ciphering key is restored via the security mode setup procedure. The option of using the restored keys allows a linkability attack on the GUTI reallocation procedure [32, 30]. As the $GUTI\_Reallocation\_Command$ does not contain a replay protection mechanism, an adversary is able to exploit this weakness. The adversary first captures a GUTI reallocation command. Later, when the UE has already updated its GUTI but not yet the ciphering key CK, the attacker replays the captured reallocation command. The victim UE has no way to detect this replay attack. It successfully decrypts this reallocation command and replies via a $GUTI\_Reallocation\_Complete$ message. This allows the adversary to distinguish the target UE from any other, as other UEs will not be able to decrypt the reallocation command and hence will not reply the completion message, even though in

the meantime the target UE was assigned with an updated GUTI. This results in the adversary being able to track the target user with minimal effort.

3.3.8 RRC Protocol Vulnerabilities / Misimplementations

The Radio Resource Control (RRC) protocol is used to set up and manage the radio connectivity between the UE and SN. The major functions of the RRC protocol include connection establishment and release functions, broadcast of system information, radio bearer establishment, reconfiguration and release, RRC connection mobility procedures, paging notification and release, etc. Within the protocol stack, it exists at the network (IP) layer. The RRC protocol is specified in 3GPP TS 25.331 [13] for UMTS and in 3GPP TS 36.331 [17] for LTE. In LTE, when the UE selects a cell in RRC *idle mode*, it does not validate whether the base station is authentic or fake. As a result, the UE may clamp on to a rogue base station. So far, the mobile telephony systems have focused on providing secure communication in the RRC *connected state* and security aspects in RRC *idle state* have not been considered. This vulnerability of UE to false base station attacks during the RRC *idle state* has been acknowledged as a *Key Issue* in TR 33.899 (sub-clause 5.4.3.1) [8].

The LTE RRC protocol also contains a "network information broadcast" function in which GUTIs associated with the SNs are broadcasted over the air [129]. These broadcasts are neither encrypted nor authenticated, hence can be decoded easily by an adversary. Since these broadcasts are location specific, techniques described in [97] can be exploited to reveal presence of subscribers in that specific area (a type of IMSI-probing attack, as explained in § 3.3.2). Another type of RRC message which contains subscriber-specific sensitive information is the "UE measurement report". In particular, two types of UE measurement reports have been exploited in the literature [129] to compromise location of subscribers:

- **Measurement Report:** *Measurement report* is a necessary part of the handover procedure of LTE networks. The SN sends a "measurement configuration" message to the UE indicating what type of measurement is to be performed. In response, the UE compiles and sends the appropriate *measurement report*. The earlier LTE specifications (Version 12.5.0 of TS 36.331 and earlier) allowed transmission of these RRC messages before establishment of a security context between the UE and SN. This has been exploited to compromise the location of subscribers by decoding of the location information contained

within these messages [129, 61]. However, later the specification was updated to allow *measurement report* transmission only after establishment of the security context between UE and SN. Although the attack descriptions in [129] mention "mapping between GUTI and IMSI via semi-passive attacks", it is unclear whether knowledge of the victim's IMSI contributes towards these attacks - hence the *property unknown* label (?) in Table 3.1.

- **Radio Link Failure (RLF) Reports:** *RLF reports* are used to troubleshoot signal coverage issues. These reports contain serving and neighboring base stations' identifiers along with their corresponding power measurements, which can be used as inputs to trilateration techniques such as [42] to determine an accurate position of the UE. The LTE standard (Appendix A.6 of [17]) does not allow transmission of *RLF reports* before establishment of a security context between the UE and SN. However, practical investigations [129] of real-world mobile networks has found that LTE phones (baseband processor to be more specific) do transmit these reports without a security context, leading to location leaks of the subscribers. This shows that the related guidelines within the standard are vague and ambiguous (described in an appendix located at the end of a 900+ page document), which leads to incorrect implementation by multiple manufacturers.

3.3.9 IMSI-based Paging

Figure 3.3 outlines the paging procedure in LTE. The Mobility Management Entity (MME) (a part of the SN's core network) is responsible for initiating paging and authentication of the mobile device, while eNodeB is the LTE base station (part of SN's access network). At the commencement of the paging, the MME starts a timer (T3413) and expects a response from the UE before the expiration of this timer. UEs in RRC *idle* state use Discontinuous Reception (DRX) also known as the *paging cycle* to reduce power consumption. This DRX cycle determines how frequently the UE checks for paging messages. The default DRX cycle is broadcast by the SN via the System Information Block (SIB). The Paging Occasion (PO) for a UE (i.e., when it wakes up to check for paging messages) is given by three numbers: the *paging cycle* $T \in \{32, 64, 128, 256\}$; the Paging Frame Index (PFI), which is an integer between 0 and $T - 1$; and a *subframe index* s where, $0 \leq s \leq 9$. The UE decodes the RRC paging messages and if it finds its identifier within this message then it initiates the acquirement of an available radio channel through the "Random Access

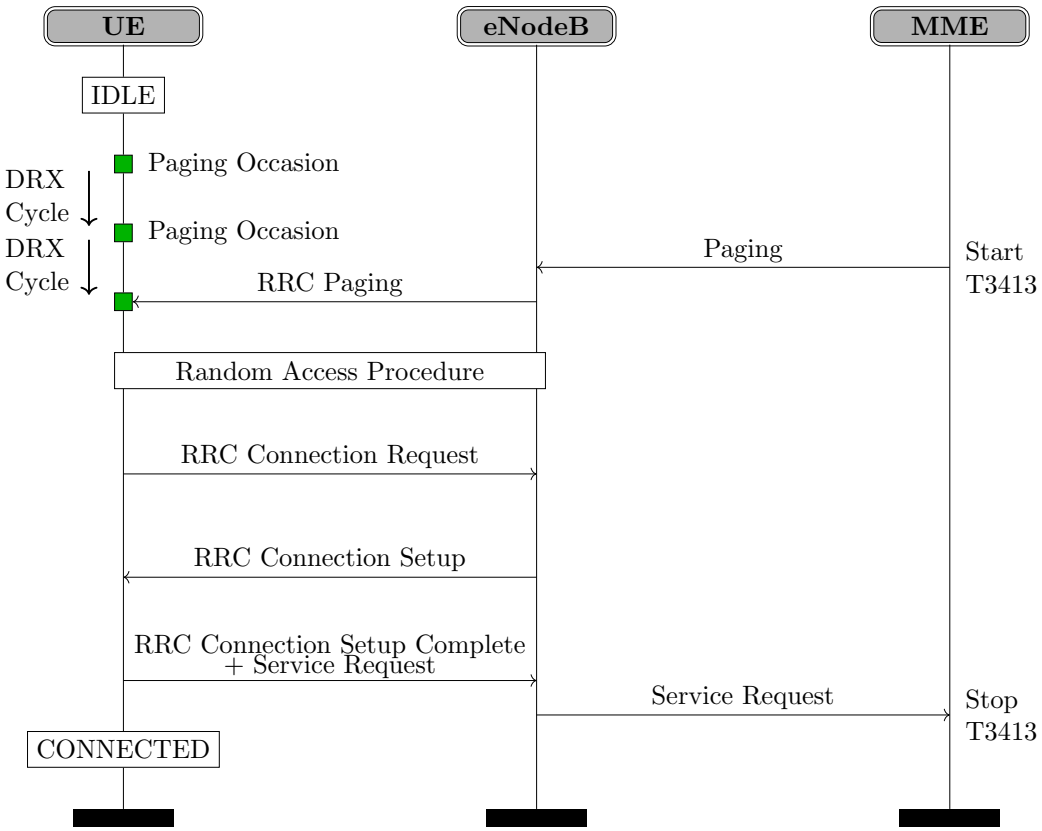


Figure 3.3: The LTE paging mechanism.

Procedure". Thereafter, the UE requests the eNodeB via the "RRC Connection Request" to configure radio resources for signalling exchange. After completion of this RRC connection setup, the UE send a "Service Request" message and enters the *connected* state.

In LTE paging, two types of identities could be used to alert idle UEs about incoming data: temporary identifier GUTI or permanent identifier IMSI. Usually, it is the GUTI which is utilized as an identifier within the paging messages. However, in situations where the SN loses its context with the UE due to a crash or restart, the provision is there to send the IMSI as the UE identifier. Using the IMSI as the UE identifier while sending paging messages has been reported as a privacy threat to users [129, 31, 132, 32].

A passive adversary can just observe the radio communication in an interested location and come to know which subscribers are located in that particular area. Since during the paging procedure a security context is not yet established between the UE and SN, an active adversary can set up a false base station in an area of interest (airports, hospitals, etc.). It can then start sending out IMSI-based paging requests

to the subscribers and, based upon the responses, will come to know which IMSIs are present in that particular area. The LTE subscribers reply to IMSI-based paging triggers via their GUTIs. Hence, this leads to a correlation between the IMSIs and GUTIs. This, combined with the initiation of paging mechanism via placing phone calls to the MSISDN (§3.3.5), allows an attacker to further correlate its IMSI and GUTI with the MSISDN. Thus active/passive listeners, fake SNs, etc. can track down subscribers with reasonable accuracy to a specific geographic area, which has serious privacy implications. IMSI-based paging has been identified as a *Key Issue* in 3GPP TR 33.899 (sub-clause 5.7.3.10) [8].

3.3.10 ToRPEDO Attack

In LTE paging, the POs are determined by the UE's IMSI. This mechanism has been exploited to verify the presence (or absence) of a target in a specific location via an attack called ToRPEDO (TRacking via Paging mEssage DistributiOn) [73]. This attack leverages the fact that the PO for a specific UE is always fixed as it is based upon its IMSI. Hence, through triggering successive paging procedures, the attacker is ultimately able to determine the presence or absence of a target UE with high confidence.

Moreover, in the ToRPEDO process, the attacker learns the last 7 bits of the UE's IMSI. We now briefly outline this leakage process. In LTE, the last 10 bits of the subcriber's IMSI are used for calculating the PO of a device. In this calculation, however, the IMSI is considered to be a 14/15-digit decimal number instead of a TBCD encoded number. Without loss of generality, if we consider $T = 128$, then successfully calculating the victim's PO will leak the last 7 bits of the victim's IMSI.

3.3.11 Linkability of AKA Failure Messages

All generations of mobile telephony suffer from a location attack known as the *Linkability of (AKA) Failure Messages* (LFM) attack [39, 31, 32]. The LFM attack exploits the fact that in an AKA protocol (see §2.7), in the event of an erroneous authentication challenge, the reason for the authentication failure is exposed to the attacker, i.e., either $MAC\_Failure$ or $Sync\_Failure$. This allows an attacker to link two different AKA sessions to identify a target user. The LFM attack is simple to execute in practice. The attacker first observes an AKA session of the target user and records the authentication challenge $(R, AUTN)$. Later, when the attacker

Table 3.2: Effect of 5G privacy enhancements upon existing attacks.

Section	5G Privacy Enhancing Features	Existing Attacks									3GPP Reference
		IMSI-catching	(Raw) IMSI-probing	GUTI Persistence	GUTI-MSISDN Mapping	GUTI Reallocation Replay	Localization via UE Reports	IMSI-paging	ToRPEDO Attack	LFM Attack	
3.4.1	SUPI Concealment	●	○	○	○	○	○	○	○	○	Sub-clause 5.2.5 of TS 33.501 [14]
3.4.2	Strict GUTI Refreshment	○	○	●	●	○	○	○	◐	○	Sub-clause 6.12.3 of TS 33.501 [14]
3.4.3	False Base Station Detection Framework	◐	○	○	○	◐	●	◐	○	◐	Annex E of TS 33.501 [14]
3.4.4	De-coupling of SUPI from Paging	○	○	○	○	○	○	●	○	○	Sub-clause 9.3.3.18 of TS 38.413 [18]
3.4.5	GUTI-based Paging Occasion	○	○	○	○	○	○	○	●	○	Sub-clause 7.1 of TS 38.304 [19]
3.4.6	Secure Radio Redirections	◐	○	○	○	◐	◐	◐	○	○	TS 38.331 [20]

Legend: ● = resolves, applicable ◐ = partial/limited effect ○ = does not resolves, not applicable

wants to check whether another AKA session belongs to the same user or not, he replays the recorded authentication challenge and observes the type of failure message received. In the case of *MAC_Failure* it is some other user, while in the case of *Sync_Failure* it is the same user. Note that in an LFM attack no further computations are required and the results are precise. Hence this is a devastating attack (albeit under additional assumptions about the attacker's capabilities) which compromises subscription location and, as an extension, allows user-traceability.

3.4 The Present - Privacy Improvements by Release 15

Release 15 comes with several new features that significantly improve subscription privacy on the radio interface [98, 123]. In this section, we review and discuss these new features. Table 3.2 provides a summary of the effect of these new features upon the vulnerabilities from previous generations.

3.4.1 Concealment of SUPI

Keeping in mind the severity of the threats posed by SUPI exposure via IMSI-catching attacks (§3.3.1), 3GPP decided to address this problem in 5G Release 15 (sub-clause 5.2.5 of TS 33.501) [14]. In the case of identification failure via a 5G-GUTI, unlike earlier generations, 5G security specifications do not allow plaintext transmissions of the SUPI over the radio interface. Instead, a public-key based privacy-preserving identifier containing the concealed SUPI is transmitted. The

public-key scheme chosen by 3GPP for this purpose is Elliptic Curve Integrated Encryption Scheme (ECIES) [131]. The concealed identifier is called SUCI. When enabled, this feature makes it infeasible for false base stations to identify or trace subscribers in a 5G-only system. The UE generates the SUCI with the public key *pk* of the HN using an ECIES-based protection scheme. This public key is securely provisioned to the UE during the USIM registration. Only the MSIN part of the SUPI is concealed by this protection scheme, while the home network identifier (MCC/MNC) is transmitted in plaintext as it is required for routing in roaming use cases.

As the *pk* comes pre-configured on the USIM, a Public-Key Infrastructure (PKI) is not needed. Also, the subscription identification is achieved in just one pass of communication, which helps in reducing the connection set-up time. Further, this scheme is oblivious to desynchronization [90] of identifiers between the UE and HN and requires simple key management, both of which lead to significant reduction in connection failures. However, there still remain aspects which require further improvement. These issues were communicated to 3GPP by European Telecommunications Standards Institute's (ETSI) Security Algorithms Group of Experts (SAGE) [59] and are discussed in further detail in §5.4.2.

3.4.2 Strict Refreshment of GUTI

In 5G Release 15 (sub-clause 6.12.3 of TS 33.501), it is mandatory to refresh the 5G-GUTI on the following occasions:

- **Initial Registration:** If the SN receives a *registration request* message of type"initial registration" or "mobility registration update" from a UE, it should send a new 5G-GUTI to the UE in the registration procedure.
- **Mobility Registration Update:** If the SN receives a *registration request* message of type "mobility registration update" from a UE, it should send a new 5G-GUTI to the UE in the registration procedure.
- **Periodic Registration Update:** If the SN receives a *registration request* message of type "periodic registration update" from a UE, it should send a new 5G-GUTI to the UE in the registration procedure.
- **Network Triggered Service Request:** Upon receiving a *service request*

message sent by the UE in response to a paging message, the SN sends a new 5G-GUTI to the UE.

These mandatory update features makes identifying or tracing subscribers based on 5G-GUTI impractical. Further, it is left to a network operator's implementation to re-assign 5G-GUTI more frequently, for example after a *service request* message from the UE not triggered by the network.

3.4.3 False Base Station Detection Framework

As evident from the description of vulnerabilities in §3.3, most attacks on previous generations leverage false base stations before the UE can go into an authenticated state. To counter such vulnerabilities, a general framework for detecting false base stations has been described in 5G Release 15 (Annex E of [14]). This network-based detection framework uses radio condition information (measurement reports of §3.3.8)) received from the devices, which can be used to make it significantly harder for false base stations to remain stealthy. The received-signal strength and location information in measurement reports can be used to detect a false base station which tries to attract the UEs by transmitting signal with higher power than that of the genuine base stations. These reports can also be used to detect a false base station which replays the original network broadcast information without any modification.

To detect a false base station which replays modified broadcast information to prevent victim UEs from switching back and forth between itself and the genuine base stations (by modifying neighboring cells, cell reselection criteria, registration timers, etc. to avoid the so called ping-pong effect), information from the broadcast information can be used to detect inconsistency from the deployment information. Further, false base stations using unusual frequencies or cell identifiers can be detected by analyzing the respective information in the received measurements reports. Networks and devices can utilise other additional security and privacy features which are proprietary to the operators. Effective false base station detection should result in significant privacy improvement. This is because it has already been proven by [101] that in case of uncorrupted mobile network participants, the AKA protocol provides anonymity guarantees to the UE.

3.4.4 Decoupling of SUPI from the Paging Mechanism

The provision of paging UE based on SUPI has been removed from 5G (sub-clause 9.3.3.18 of TS 38.413) [18]. Moreover, the calculation of the paging frame index and paging occasions is no longer based on SUPI and is instead based on 5G-GUTI. Coupled with the mandatory 5G-GUTI update mechanism (§3.4.2), this makes it infeasible for false base stations to use paging messages for identifying or tracing subscribers.

3.4.5 GUTI-based Paging Occasions

While, in LTE, POs were determined based on the IMSI, in 5G they are based on a temporary identifier (called 5G-S-TMSI) which is a subset of the GUTI. The result of this change is that now the ToRPEDO attack (§3.3.10), which leveraged fixed POs for a target UE, is no longer able to exploit the permanency in paging timings. This enhancement, along with frequent GUTI refreshment (§3.4.2), results in enhanced user privacy.

3.4.6 Secure Radio Redirections

It is mandatory in 5G Release 15 (TS 38.331 [20]) to integrity protect RRC messages that redirect devices. This feature makes it infeasible for false base stations to perform rogue redirections. As a result, the level of difficulty to launch various privacy attacks which rely on rogue redirections increases manifold.

3.5 The Future - Outstanding Issues, New Attacks and Proposed Measures

The successful deployment of future 5G systems requires resolution of the outstanding subscription privacy issues. In this section, we highlight the subscription privacy vulnerabilities which were not addressed by Release 15. We also discuss the recent literature which either suggests improvements or presents new attacks on 5G subscription privacy.

3.5.1 Unresolved Vulnerabilities

An examination of Table 3.2 reveals that there are three privacy issues from previous generations which were not aptly addressed by Release 15: (Raw) IMSI-probing (§3.3.2), C-RNTI-based tracking (Section 3.3.6) and the AKA-protocol based LFM attack (§3.3.11). Regarding (Raw) IMSI-probing, as already discussed in §3.3.2, it is highly unlikely that 3GPP will adopt countermeasures to this particular problem because of the overhead of the required dummy signalling. The only feasible solution to handle the C-RNTI-based privacy breaches is to employ a network-wide PKI [79] since this requires encryption of these pre-authentication identifiers. This is unlikely to be a desirable option for 3GPP due to the high costs associated with deploying and maintaining a PKI.

As regards the LFM attack, Arapinis et al. [31], while highlighting this vulnerability, also proposed a fix to resolve this problem. The proposed fix requires the HNs to have a public/private key pair, where each USIM stores the public key of its HN. The AKA failure messages are then encrypted using the network's public key. They verified the privacy properties of their fixes using the automated symbolic analysis tool `ProVerif` [35]. However, their proposed fix has been shown by Fouque et al. [62] to be still plagued with certain privacy weaknesses. Fouque et al. presented their own improved variant of the public-key based fix for the LFM vulnerability. However, the solution of Fouque et al. has been shown by [94] to be vulnerable to permanent desynchronization attacks. 3GPP has never considered adoption of these proposals, essentially because they are public-key based and introduce significant overhead. As the UE and HN already share common secrets between them, the better way forward seems to resolve this issue via symmetric key solutions. We explore such approaches further in §3.5.3.

3.5.2 New Attacks on 5G Subscription Privacy

Recently, Borgaonkar et al. [38] have presented new attacks against all variants of the AKA protocol, including 5G-AKA, which breach subscribers' privacy. These attacks exploit a logical vulnerability in the AKA protocol's failure mechanism. The vulnerability stems from the use of XOR within the re-sync token $AUTS$ (see Figure 2.3), which is concatenation of two parameters: $CONC^*$ and MAC^*. Based upon this logical vulnerability, [38] presented two attacks against 5G user privacy: Activity Monitoring Attack (AMA) and Location Confidentiality Attack (LCA).

In AMA, the adversary tries to learn the n least-significant bits of SQN_{UE} at two different time instances, t_1 and t_2. Thereafter, from the difference between the sequence numbers (corresponding to successful authentication sessions), the attacker infers the volume of "activity" (number of calls, SMSs, etc) a particular user has performed between these two time instances - hence the name Activity Monitoring Attack. In LCA, the aim of the attacker is to find out whether some targeted UE is present in a certain location or not. We analyze these attacks in more detail for their effectiveness, practicability and potency against 5G in Chapter 4.

3.5.3 Fixing LFM, AMA and LCA

As discussed previously in §3.5.1, a symmetric-key based solution is required which should together resolve the three vulnerabilities of LFM, AMA and LCA. We now briefly review some of these solutions proposed by [38].

3.5.3.1 Symmetrically Encrypting SQN_{UE} (Fix_1)

This fix consists of modifying the sequence number concealing mechanism. Instead of using XOR to conceal SQN_{UE}, this fix utilizes symmetric encryption. The resulting fix is depicted in Figure 3.4a. To counter the LFM attack, it suffices to hide the reason for the 5G-AKA protocol failure inside the ciphertext $CONC^*$. The authors of [38] claim that this fix is easy to deploy in the current cellular system as it only requires changes in the baseband module of the UE (i.e. ME) and not USIM. This seems strange as it is the USIM (not the mobile handset) which is directly under the control of the mobile network operator. This solution suffers from a flaw: when an attacker triggers a failure message by injecting the same authentication challenge twice while the SQN_{UE} has not being updated in the UE, then the replied $CONC^*$ will be the same as before, leaking to the attacker that SQN_{UE} is unchanged.

3.5.3.2 Correctly Randomizing $AUTS$ (Fix_2)

Another way to fix the AMA and LCA is to generate a new random ($RAND^*$) to conceal SQN_{UE} instead of utilizing the one ($RAND$) received in the authentication challenge. This new random $RAND^*$ needs to be sent back in clear to the HN along with $AUTS$ for decryption of SQN_{UE}. Figure 3.4b depicts this solution. Note that the original $RAND$ must be used in calculation of MAC^* to guarantee a fresh

response to the received authentication challenge. Otherwise, an attacker will be able to replay an old response back to the HN, forcing it to synchronize its SQN_{HN} to an older value. Also note that this fix does not resolves LFM attack on its own.

3.5.3.3 Combining Fix_1 and Fix_2 (Fix_3)

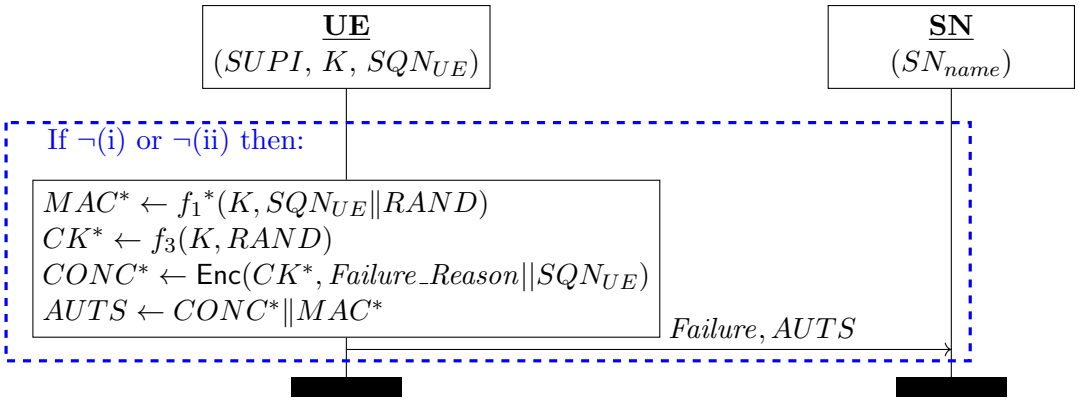
Both Fix_1 and Fix_2 have limitations of their own. Fix_1 suffers from a minor flaw, while Fix_2 is not applicable for LFM attack. For a comprehensive solution, which resolves both of these issues, we combine Fix_1 and Fix_2 as suggested in [38]. This combined fix is depicted in Figure 3.4c and addresses LFM, AMA and LCA without any known flaws / limitations.

3.5.4 Quantum-secure and Downgrade-resistant SUPI Protection

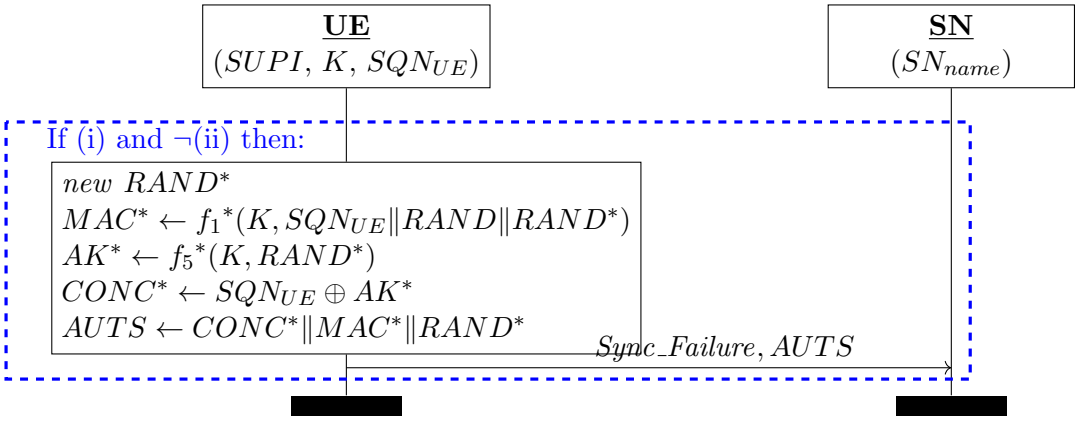
As pointed out by [109], the current ECIES-based SUPI protection solution is vulnerable to quantum cryptanalysis. Until the publication of the 3GPP's public-key based protection mechanism, the technical problem of finding a SUPI protection solution remained opened in a purely symmetric-key setting. However, later on, we proposed a solution for SUPI protection that works entirely within symmetric-key domain. This solution was presented at the 2018 Security Standardization Research (SSR) conference [83] and addresses all the shortcomings of the ECIES-based mechanism. We discuss this alternative proposal in further detail in Chapter 5. Interestingly, another paper [89] presented at the same venue proposed a protection mechanism for the downgrade attacks against 5G-AKA. These two solutions can be combined together to come up with a 5G SUPI protection mechanism which is both quantum-secure and downgrade-resistant. We present this combined solution in Chapter 6.

3.5.5 IBE-based SUPI Protection

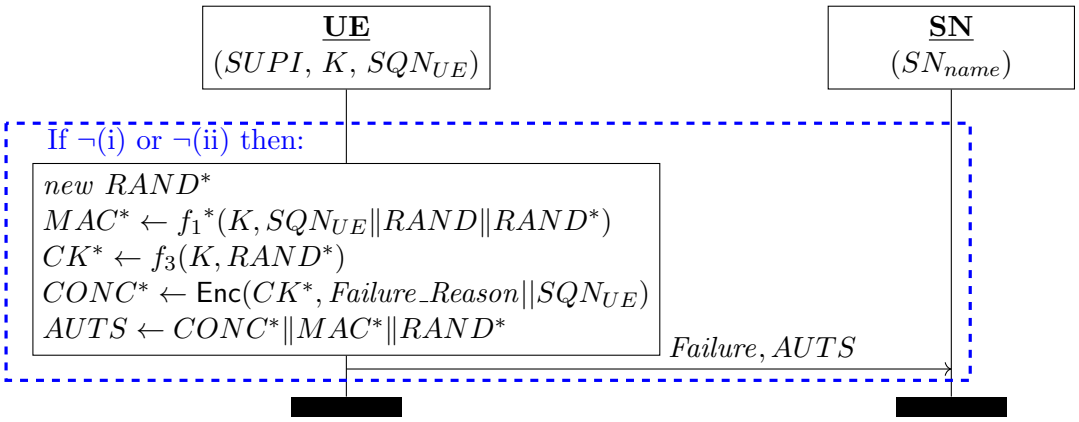
Both the current 3GPP SUPI protection mechanism (§3.4.1) and our alternative symmetric-key proposal (detailed in Chapter 5) hide only the MSIN part of the SUPI, while the MCC and MNC part is sent in clear over-the-air to the SN for routing of the SUCI to the correct HN. Also, to increase look-up efficiency, mobile network operators divide their subscriber database into further sub-domains [140]. Therefore, it is required that the SUCI be delivered to the correct sub-domain within the HN. Typically, this requires between one and three digits after the MCC/MNC



(a) Fix_1: Symmetrically encrypting SQN_{UE}.



(b) Fix_2: Correctly randomizing $AUTS$.



(c) Fix_3: Combining Fix_1 and Fix_2.

Figure 3.4: Proposed fixes for 5G-AKA failure messages.

in the MSIN to be sent in clear as part of the routing information [141]. All this results in weakening of the privacy protection being offered to the mobile subscriber as a significant part of its identity is now exposed to an attacker.

Another limitation of the 3GPP protection mechanism and our alternative proposal is that the SN is entirely dependent upon the HN for revelation of the SUCI and the associated LI purposes [92]. Several countermeasures have been proposed in 3GPP meetings for handling of this issue [43, 58, 95, 115, 116]. All of these suggested countermeasures introduce overhead either due to additional signalling messages or due to requirement of new parameters. Moreover, there is nothing stopping the UE and its HN from colluding to provide the SN with a false SUPI.

To counter these limitations, Khan and Niemi [91] proposed a 5G-SUPI protection scheme based on Identity-based Encryption (IBE). In this scheme, the UE's HN act as the Private Key Generator (PKG). IBE-based schemes inherently resolve the exposure of partial MSIN and provide better LI guarantees as the SN can now work out the SUPI from the SUCI independently of the HN. The proposal by [91] can be argued to be a better alternative to the current 3GPP mechanism, though the associated key-revocation is quite complex. However, compared to our solution ([83]), it is not quantum-secure and the increase in computational and signalling overhead is much higher. Also, it is unclear whether the IBE-based solution can be used in combination with the downgrade protection proposal of [89]. Given these limitations, in the long-term, our solution seems more preferable.

3.5.6 Study on Protection against False Base Stations

Another important avenue which still requires further research is that of protection against false base station attacks. Though 5G Release 15 provides a false base station detection framework (§3.4.3), its status as of now is informative only. Moreover, the provided framework is generic in nature and focuses only on the detection aspects. Very recently, 3GPP has initiated a comprehensive study [23] which focuses on security enhancements against false base stations for the next 5G Release 16. The aim is to study the potential threats and privacy issues associated with false base station scenarios and identify potential solutions for mitigating the risks caused by false base stations. As various attacks against 5G subscription privacy on the radio interface exploit false base station as the underlying platform, this study will also contribute towards subscription privacy enhancement in Release 16.

Table 3.3: Important recent survey publications related to 5G security and privacy.

Reference	Publication Year	Application Area	Main Contribution	Relevance to 5G Privacy
[126]	2018	2G, 3G, 4G	A survey of existing literature on attacks in previous generations (GSM/UMTS/LTE) of mobile telephony.	Suggests research directions / improvements for 5G subscription privacy.
[135]	2018	ICN	A survey about security, privacy, and access control in information-centric networking.	The privacy attack scenarios discussed are also applicable to 5G networking concepts.
[28], [29]	2017/2018	5G	An overview of 5G security challenges and solutions.	Discusses the privacy challenges in 5G from the user's perspective.
[93]	2019	5G	A survey on the security and privacy of 5G.	Focused on portraying a landscape of futuristic security threats to 5G.
[78]	2019	5G	A survey of remaining security and privacy issues in 5G.	Proposes PKI integration to resolve outstanding issues.
[60]	2018	4G, 5G	A survey of existing authentication and privacy-preserving schemes for 4G and 5G cellular networks.	Discusses privacy attacks on 5G networks and provides recommendations for further research.
[65]	2017	5G	A survey on green communication and the associated security challenges in 5G networks.	Reviews privacy aspects of various 5G enabling technologies like machine-to-machine (M2M) communications, etc.
[124]	2017	SDN	A survey of issues and challenges in designing SDN based 5G networks.	No explicit focus on 5G privacy rather provides SDN based security solutions for 4G and 5G networks.
[48]	2019	5G	A survey on the security of alternative computing paradigms for 5G networks.	Emphasizes the applicability of alternative computing paradigms for enhancement of subscriber privacy.

3.6 Related Work

We believe there does not exist any prior work in the published literature with exclusive focus on 5G subscription privacy. The probable reason for this seems to be that 5G is a very nascent technology within which extensive development and upgrades were undertaken as late as June 2019. Table 3.3 presents a summary of the related literature which has considered security and privacy in 5G or 5G-like networks. Here, we briefly discuss the work carried out in these publications.

Rupprecht et al. [126] categorized and systematized attacks in existing mobile generations (GSM/UMTS/LTE) by their aim, impact and attacker capabilities. They further identified future research directions for 5G networks based on these existing security and privacy issues. The main difference between [126] and our work is that we also consider 5G Release 15, while the privacy analysis of [126] is limited only to the previous generations. Tourani et al. [135] have analyzed security, privacy and access control within the scope of Information-centric Networking (ICN). ICN is a

networking paradigm which focuses on content of the traffic rather than its origin - a concept similar [125] to *network slicing*[3] in 5G. Ahmad et al. [28, 29] analyzed generic security and privacy threats to 5G networks and suggested possible solutions to these threats from the published literature. As both of these works were carried out before the publication of the 5G standard, they are mostly speculative in nature. Khan et al. [93] have presented a survey about security and privacy of 5G. The 5G privacy issues discussed in [93] are again speculative in nature as the manuscript was drafted before the publication of 5G Release 15.

Jover [78] discussed security challenges faced by 5G. The main focus of this work was the integration of a Public Key Infrastructure (PKI) within the current 5G network architecture to resolve outstanding security and privacy issues. Ferrag et al. [60] presented a survey of existing authentication and privacy-preserving schemes for LTE and 5G mobile networks. They provided a classification of threat models in 4G and 5G cellular networks in four categories: attacks against privacy, attacks against integrity, attacks against availability, and attacks against authentication. They also provided a classification of the respective countermeasures into three categories: cryptographic methods, humans factors, and intrusion detection methods. It seems that the work of [60] presumed that all the analysis and contextualization with respect to 4G can be seamlessly applied to 5G. The reason for this is because at the time of publication of [60] (January, 2018) even the Stage-2[4] of 5G Release 15 was not completed (see Figure 3.1).

Gandotra and Jha [65] presented a survey on various energy-efficient scenarios for green communication in 5G and the related security aspects. For improving the battery lifetime of user terminals, [65] proposed transmitting information through relays and discussed security susceptibilities via these relays and the associated countermeasures. However, [65] did not consider 5G privacy. Rangisetti and Tamma [124] explored the aspects related to migration of mobile network infrastructure in LTE and 5G to Software Defined Networking (SDN) and Network Function Virtualization (NFV). It further elaborated security issues in migration to these new technologies and suggested SDN-based solutions. The work by [124] is focused on the security issues during architecture migration and not on subscription privacy. Choudhry

[3] Network slicing is a form of virtual network architecture which delivers greater network flexibility by allowing traditional network architectures to be partitioned into virtual elements that can be linked through software.

[4] "Stage-2" is a stage where logical analysis, devising an abstract architecture of functional elements and the information flows amongst them across reference points between functional entities is carried out.

and Sharma [48] surveyed recent computing paradigms as alternative mechanisms for the enhancement of 5G security. This work particularly focuses on the feasibility of catalytic and osmotic computing in 5G networks and not subscription privacy.

3.7 Chapter Summary

Although 5G offers better privacy guarantees than its predecessors, this work showed that there still remain significant issues which need rectifying. Several privacy vulnerabilities that remain unresolved in 5G Release 15 were highlighted. To address the identified privacy gaps, this chapter also proposed improvements for future versions of the 5G standard. The study concludes that new and more rigorous privacy protection mechanisms are required to guarantee subscription privacy in 5G. In particular, for a quantum-secure future, 3GPP should consider accompanying the current subscriber identification protection mechanism (being the only public-key based mechanism in 5G) with the symmetric proposal of [83].

Chapter 4

Efficacy of New Privacy Attacks against 5G-AKA

This chapter provides an analysis of the attacks on 5G privacy by Borgaonkar et al. [38]. We evaluate these attacks for their effectiveness, practicability and potency against 5G. This analysis was published at the International Conference on Security and Cryptography 2019 [84].

4.1 Introduction

The 3GPP standard for 3G/4G mobile telephony security [16] provisions an Authenticated Key Agreement (AKA) protocol for establishment of a secure channel between mobile subscribers and their service providers. The AKA protocol for 4G networks is similar to that of 3G with slight differences in identifier and key management. For 5G, an enhanced version of this AKA protocol called the 5G-AKA was introduced by 3GPP [14]. Apart from typical security requirements, an important consideration for these AKA protocols (and their associated mechanisms) is end-user privacy.

Borgaonkar et al. [38] have revealed a new logical vulnerability in one of the associated mechanisms (sequence number re-synchronization) of the 3G/4G/5G AKA protocols. Based upon this vulnerability, they have presented two privacy attacks: Activity Monitoring Attack (AMA) and Location Confidentiality Attack (LCA). The AMA allows an attacker to learn subscribers' mobile service consumption patterns while the LCA allows tracking of mobile subscribers, thus breaking location confidentiality. Borgaonkar et al. claim that these attacks adversely affect all generations of mobile telecommunications, including 5G. More importantly, they state that these attacks have been acknowledged by the requisite standardization bodies and that remedial actions are underway for the future 5G specifications.

In this chapter, we analyze the efficacy of these new privacy attacks against 5G. The reason for confining this analysis to 5G is due to the fact that relatively facile attacks (IMSI-catching [63], Linkability via Failure Messages (LFM) [31][1]) provisioning more disastrous breaches of subscriber privacy already exist for the previous generations (2G/3G/4G). Effective countermeasures for these existing attacks were not incorporated in the already-deployed standards because of the high upgrade costs involved. It is thus too late to propose any amendments for the 2G/3G/4G specifications. The findings of our analysis contradict some of the claims made in [38]. Specifically, we show the following:

- The AMA is infeasible to execute in 5G networks.
- The LCA is a direct extension of an existing privacy vulnerability [31] that exploits linkability of the AKA failure messages. Moreover, we demonstrate that the results obtained with this extension attack are less effective than those achieved via the existing vulnerability.
- Contrary to [38] which claims dedicated fixes are required for their attacks, we establish that in case of effectual countermeasures introduced against the existing vulnerability of [31], both AMA and LCA will be rendered futile.
- The associated security and privacy analysis of the modified 5G-AKA in [38], carried out in a symbolic model, is inaccurate and error prone due to omission of important aspects specified within the 3GPP standard.

The rest of the chapter is organized as follows: §4.2 provides the requisite background, §4.3 details the logical vulnerability in the 5G-AKA, AMA and LCA are described in §4.4 and §4.5, respectively, §4.6 analyzes the attacks and §4.7 provides concluding remarks.

4.2 The 5G-AKA

Before considering the details of the privacy attacks, we outline the 5G-AKA upon which these attacks are based. Though §2.7 already details the 5G-AKA, we restate the pertinent aspects here for continuity purpose. Figure 2.3 shows details of the 5G-AKA and its associated failure mechanisms. In Figure 2.3 functions f_1,..., f_5,

[1]This linkability attack is also valid for 5G Release 15.

f_1^* and f_5^* are unrelated symmetric key algorithms, f_1, f_2 and f_1^* act as message authentication functions, while f_3, f_4, f_5 and f_5^* are used as key derivation functions. Key derivation is performed using the Key Derivation Function (KDF) specified in 3GPP TS 33.220 [11] and SN_{name} is the global identity of the SN. A successful 5G-AKA culminates in the derivation of the anchor key K_{SEAF} by the SN and UE from which further keys for securing various layers of communication are derived. The two cases of authentication failure for the 5G-AKA are as follows:

1. **MAC_Failure:** As the first step in authentication confirmation, the UE checks whether the received MAC value is correct or not. In case of a failure [Case ¬(i) in Figure 2.3], the UE replies with a *MAC_Failure* message back to the SN.

2. **Sync_Failure:** After MAC verification, the UE checks the freshness of the sequence number SQN received in the authentication challenge. In case of this failure [Case (i) and ¬(ii) in Figure 2.3], it responds with a *Sync_Failure* message along with a re-sync token $AUTS$. Note that in Figure 2.3, the sequence number freshness check is denoted by $XSQN_{HN} > SQN_{UE} - \triangle$. What this means is that there is some "window" of size $\triangle$, within which sequence numbers smaller than the current sequence number of UE will be accepted given they previously had not been received by the UE. This mechanism is there to handle out-of-order delivery of challenge messages from HN to UE. We discuss this aspect in further detail in Section 4.6.3.

In addition to the requirements of mutual authentication and data confidentiality, it is crucial that SQN is protected from an eavesdropper during the establishment of a secure channel between the UE and SN as its exposure may lead to the compromise of the identity and location of a user [9].

4.3 The Logical Vulnerability

The logical vulnerability of [38] affecting user privacy stems from the use of XOR within the re-sync token $AUTS$, which is concatenation of two parameters: $CONC^*$ and MAC^*. The parameter $CONC^*$ contains the current sequence number of the UE in a masked form as $SQN_{UE} \oplus AK^*$, where $AK^* = f_5^*(K, RAND)$. Note that during calculation of the masking key AK^*, the value $RAND$ is extracted from the received authentication challenge. Hence, in the case of receiving the same

authentication challenge twice at two different times t_1 and t_2, the masked sequence numbers in their corresponding $AUTS$ tokens will be:

$$CONC_1^* = SQN_{UE}^1 \oplus AK_1^*, \ where \ AK_1^* = {f_5}^*(K, RAND)$$
$$CONC_2^* = SQN_{UE}^2 \oplus AK_2^*, \ where \ AK_2^* = {f_5}^*(K, RAND),$$

where SQN_{UE}^1 is the sequence number of UE at time t_1 and SQN_{UE}^2 is the sequence number at time t_2. Therefore, the adversary can compute:

$$CONC_1^* \oplus CONC_2^* = SQN_{UE}^1 \oplus SQN_{UE}^2.$$

Next we detail the two attacks presented in [38] which, by exploiting this vulnerability, try to compromise user privacy.

4.4 Activity Monitoring Attack

In this attack the adversary tries to learn the n least significant bits of SQN_{UE} at two different time instances, t_1 and t_2. Thereafter, from the difference between the sequence numbers (corresponding to successful authentication sessions), the attacker infers the volume of "activity" (number of call, SMS, etc) a particular user has performed between these two time instances, hence the name *Activity Monitoring Attack*. As we will see shortly, to mount this attack the adversary requires malicious interaction with both UE and HN (via SN). Hence, the compromise of both *identity confidentiality* and *location confidentiality* of the target UE are prerequisites to launch an AMA.

Details of a single instance of the attack at a particular time t are now explained. The online phase of the AMA is depicted in Figure 4.1. During this phase the attacker first fetches $2^{n-1} + 1$ successive authentication challenges from the SN for the targeted UE. The attacker then sends a particular $n + 1$ of these challenges to the UE, each followed by a replay instance of the initially received authentication challenge $(RAND_0, AUTN_0)$, and records the corresponding $n + 1$ resync tokens; i.e. $AUTS^{'}$ and $AUTS_j$ $(for \ j = 0 \ to \ n - 1)$.

In the offline phase, utilizing the logical vulnerability as elaborated earlier in Section 4.3, the attacker retrieves the following values from the recorded resync tokens:

$$\delta_i = SQN_{HN}^0 \oplus (SQN_{HN}^0 + 2^i) \quad for \ 0 \leq i \leq n - 1,$$

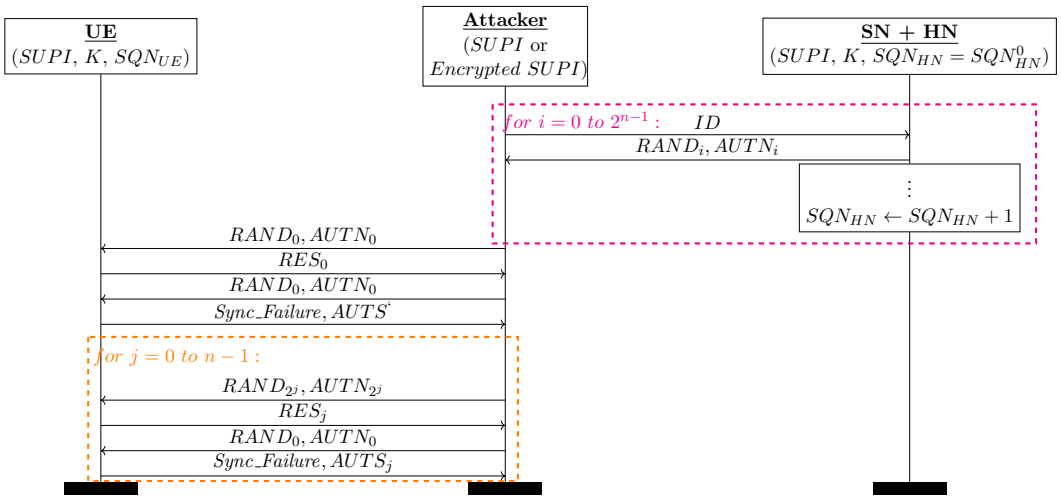


Figure 4.1: The online phase of the AMA.

Data: δ_i *for* $0 \leq i \leq n-1$
Result: $X = n$ least significant bits of SQN^0_{HN}
$X \leftarrow [0, 0, \ldots, 0]$ `// init an array of size` n
for $i \leftarrow 0$ **to** $n-1$ **do**
  `// Analyze` δ_i `at bit positions` i, $i+1$
  $(b_1, b_2) \leftarrow (\delta_i[i],\ \delta_i[i+1])$
  if $(b_1, b_2) \Leftrightarrow (1, 0)$ **then**
    `// No remainder propagates when` $SQN^0_{HN} + 2^i$
    $X[i] \leftarrow 0$
  else if $(b_1, b_2) \Leftrightarrow (1, 1)$ **then**
    `// A remainder propagates when` $SQN^0_{HN} + 2^i$
    $X[i] \leftarrow 1$
  else
    `// Not possible`
    Error
  end
end
return (X)

Figure 4.2: SQN inference algorithm.

where SQN^0_{HN} is the initial value of the SN's sequence number at the start of the attack. Note that due to receipt of the first authentication challenge $(RAND_0, AUTN_0)$ from the adversary, the UE will also sync its sequence number to this value at the start of the attack. Further, by feeding these n values into *SQN Inference Algorithm* (Figure 4.2), the attacker extracts the n least significant bits of SQN^0_{HN}.

4.5 Location Confidentiality Attack

As another consequence of the logical vulnerability of Section 4.3, [38] presented a Location Confidentiality Attack (LCA); i.e. finding out whether some targeted UE is present in a certain location. Note that we present LCA as explained in [38]. We claim there are several erroneous assumptions upon which this attack is based and we will highlight these when we undertake the corresponding analysis in Section 4.6.2. The LCA proceeds as follows:

1. The attacker observes a 5G-AKA session of some targeted user[2] UE_x and extracts the corresponding $CONC_x^*$ value by replaying the observed authentication challenge to UE_x.
2. After some time, if the attacker wishes to check whether another unknown 5G-AKA session belongs to UE_x or not, the attacker again replays the earlier observed challenge from Step (1) to this unknown user and obtains $CONC_?^*$.
3. Now based upon the value $CONC_x^* \oplus CONC_?^*$, the attacker can infer (with non-negligible probability) whether this new user is UE_x or not. In the case of some other user this will be a random value, while in the case of UE_x it will equate $SQN_{UE_x}^{old} \oplus SQN_{UE_x}^{current}$ due to canceling out of the common masking key AK^*. This value (dependent upon the lapsed time) should be small in the case of user UE_x.

4.6 Analysis

4.6.1 Analysis of AMA

4.6.1.1 Infeasible Prerequisites

As elaborated earlier in Section 4.4, to launch an AMA the adversary first needs to compromise the target's identity and location confidentiality. While such a compromise is easy to manage in earlier generations (3G/4G) via IMSI-catching attacks [63], how this will be achieved in 5G is not clear. With a randomized public-key encryption mechanism in place to protect direct exposure of the $SUPI$ during the *identification phase*, such a compromise is highly unlikely in 5G Release 15. In §5.2

[2]Note that it is not necessary for the attacker to know the $SUPI$ of the user to launch this attack.

of [38], in the case of an unknown $SUPI$, the use of $SUCI$ (the randomized encryption of $SUPI$) is suggested for fetching the requisite authentication challenges from the SN. This would require correlating the $SUCI$ to the appropriate $SUPI$, which in the case of a secure encryption scheme is not possible. The most convincing implementation of AMA in 5G would look something like this: the attacker follows the victim closely[3] and observes the victim's attach procedure (utilizing $SUCI$) to the network. We stress that all this needs to be undertaken in isolation without the presence of other mobile subscribers in the concerned attack area. Such requirement of physical tracking of the target in AMA render it unattractive for its automated use in 5G.

The prospect of the repeated use of $SUCI$ for fetching of successive authentication challenges from the SN to launch the AMA is possible because the current identification mechanism in 5G [14] is susceptible to replay attacks. No dedicated replay prevention mechanism has been built into the 5G randomized encryption scheme used for $SUPI$ protection. This was highlighted to 3GPP by the European Telecommunications Standards Institute Security Algorithms Group of Experts (ETSI SAGE) during their evaluation of the $SUPI$ protection mechanism for 5G [59]. We propose an alternative $SUPI$ protection mechanism for 5G (detailed in Chapter 5) which prevents such $SUCI$ replay attacks [83]. We stress that adoption of such a mechanism will render attacks such as AMA infeasible.

4.6.1.2 Requesting Batch of AVs

Unlike the previous generations, Clause 6.1.3.2.0 of [14] does not support requests for issuing multiple Authentication Vectors (AVs) for 5G-AKA. Also, after issuance of each 5G AV, the HN waits for a response from the SN after successful mutual authentication and key agreement between UE and SN, as elaborated in Figure 2.3. Hence, the adversary has to wait for the expiration of the timeout of the currently issued AV before the next AV issuance request can be entertained by the SN. This considerably increases the time complexity of AMA's online phase in 5G.

[3] In this case the *identity confidentiality* and *location confidentiality* are already compromised as the attacker can already identify the target and is aware of its location.

4.6.1.3 The Accuracy of AMA Assumptions

Essentially AMA tries to reveal the n least significant bits of SQN_{UE} at two different time frames for the target user. Thereafter, based upon the assumption that each sequence number increment corresponds to a successful AKA session, it deduces that the difference between the two SQN_{UE} reveals the user's service consumption during that time interval. The problem with this assumption is that the difference between the sequence numbers does not "fully" corresponds to successful AKA sessions. Many times, due to network failure or channel noise (bad weather, etc), legitimate messages may get lost during transmission and may not reach the intended destination. On the other hand, it may be the case that a user is genuinely under attack by some active attacker. In such cases, the end result would be the non-utilization of the affected sequence numbers. Thus, while a difference in sequence numbers may give a rough idea about the user's service consumption, its efficacy is dependent upon many other factors.

Another assumption that adversely affects AMA's accuracy is the inference of SQN_{UE} from SQN_{HN}. Note that at the start of the AMA, SQN_{UE} is forced to update to the value SQN^0_{HN}, the initial value of the SQN_{HN}. The presumption behind this step is that the two values should be equivalent, which may or may not be the case. It is quite possible (due to a variety of circumstances) for SQN_{HN} to be much higher than that of SQN_{UE} at the start of AMA. In such scenarios, AMA's accuracy about the target's activity is negatively impacted.

4.6.1.4 Severity of AMA

In [38], it is claimed that AMA breaches subscribers' privacy more severely than either *location confidentiality* or *identity confidentiality* attacks. This seems to be an overstatement as compromise of the permanent identity or location is arguably a more severe breach of privacy than the exposure of a number of voice calls or SMSs sent by a user. Otherwise, such a breach of privacy would have been mentioned in the official 3GPP mobile subscribers' privacy requirements [9]. In fact, breach of a user's identity and location does not only violate the user's privacy but can lead to physical attacks. For (a sensational) example, consider the scenario where a bomb explosion is triggered automatically when a high value target's presence is detected in the near vicinity of an *IMSI-catcher* [66, 36].

4.6.2 Analysis of LCA

4.6.2.1 No Activity Monitoring

Unlike AMA, LCA does not presume any prerequisite compromises (such as identity and location confidentiality) about the target which makes it a much easier attack to launch in practice. Moreover, LCA targets location confidentiality of a user instead of its service consumption, which is a more severe breach of privacy as discussed in §4.6.1. In a way, LCA can be considered as a more direct application of the logical vulnerability of §4.3. Though [38] presented LCA as an extension of their primary attack AMA, we argue that LCA is a much more significant attack than AMA as it does not require fetching of authentication vectors from the SN, nor running of the SQN Inference Algorithm, and is simple to execute. However, we stress that there is no *activity monitoring* (contrary to the claim made in the Footnote No. 2 of [38]). This is because, now, what the attacker gets after a successful LCA is:

$$CONC^{*}_{UE} \oplus CONC^{*}_{?} = SQN^{old}_{UE} \oplus SQN^{current}_{UE}.$$

Note that the presumption for this is that the value $SQN^{old}_{UE} \oplus SQN^{current}_{UE}$ will be small (less than some threshold value). So there are two aspects which hinder the accurate inference of *activity monitoring*:

1. The attacker is already operating the LCA under the presumption of a small increase in SQN_{UE}, which renders the aspect of *activity monitoring* ineffective.
2. Unlike AMA, in LCA the attacker is unable to extract the n least significant bits of SQN_{UE}. What the attacker actually gets are the positions of the bits of SQN_{UE} which flipped their value (either 0 to 1 or 1 to 0), hindering an accurate estimate of the difference between the two values. Nevertheless, there is some leakage from a cryptographic viewpoint.

4.6.2.2 No Requirement of Dedicated Fixes

Having established that LCA is not another version of AMA but, rather, an attack targeting location confidentiality in its own right, we turn our attention to another important dimension. All generations of mobile telephony (including 5G Release 15) suffer from an existing location attack known in the literature as Linkability of Failure Messages (LFM) attack [31] (discussed previously in §3.3.11). The LFM attack

exploits the fact that in the case of an erroneous authentication challenge, the reason of the authentication failure is exposed to the attacker; i.e. either *MAC_Failure* or *Sync_Failure*. This allows an attacker to link two failure messages together to identify a target user. LFM is much simpler to execute than LCA. In LFM, the attacker first observes an AKA session of the target user and records the authentication challenge $(RAND, AUTN)$. Later, when the attacker wants to check whether another AKA session belongs to the same user or not, he replays the recorded authentication challenge and observes the type of failure message received. In the case of *MAC_Failure* it is some other user, while in the case of *Sync_Failure* it is the same user. Note that in LFM, unlike LCA, no further computations are required and the results are precise. Hence, it is a more devastating attack than LCA.

In [38] it is claimed that LCA will work even if LFM attack gets patched. The reason behind this claim seems to be the (erroneous) assumption that a countermeasure for the LFM attack will only hide the reason of authentication failure and not the rest of the failure message contents (including $AUTS$ token) leading to the logical vulnerability of Section 4.3. However, the solutions in the literature proposing countermeasures to the LFM attack suggest otherwise. This is essentially because the indistinguishability experiments proving unlinkability in these solutions cover all aspects of unlinkability and not only the reason of the authentication failure. As a concrete example, we consider the countermeasure proposed in [31]. In case of an authentication failure (due to any reason), the whole failure message including the resync token is encrypted by the network public key. Hence, the logical vulnerability of §4.3 gets resolved before it can be exploited. This leads us to the deduction that, in reality, LCA is a more complex version of the LFM attack. Surprisingly, in 5G Release 15 no countermeasures for this potent LFM attack have been adopted. Though the authors of [38] present LCA as a distinct attack from the LFM attack, suggesting that dedicated countermeasures independent of existing attacks would be required, it is not hard to see that a suitable countermeasure (as already suggested in §3.5.3) against the LFM attack will also render both AMA and LCA ineffective. This is because now the attacker will not be able to the exploit the resync tokens $AUTS$ to launch AMA or LCA.

4.6.3 The Curious Case of Out-of-Order Message Delivery

Although there have been a number of formal analyses of the 5G-AKA [33, 49] in the *symbolic model* using tools such as Tamarin Prover [106], and 3GPP has been using

this approach for protocol evaluations [5], the problem has always been the necessary abstraction required during the transformation from the real-world conditions to the underlying mathematical model of the system being evaluated. As a concrete example, consider the case of the analysis carried out in [5]. Even after formal analysis, a number of vulnerabilities were later discovered in the 3G-AKA. Another example is that of [33], whose analysis of the 5G-AKA failed to capture the privacy flaws pointed out in [38]. While the formal analysis of 5G-AKA undertaken in [38] is based upon enhanced system models which consider the $AUTS$ tokens of the *Sync_Failure* messages, there is an important aspect which was missed; i.e. how the 5G-AKA (and the earlier AKA protocols) handle out-of-order delivery of the authentication challenges from the SN to UE.

As per 3GPP specifications [9], the mechanism in the UE for verifying the freshness of sequence numbers should to some extent allow the out-of-order delivery of sequence numbers. This is to ensure that the authentication failure rate due to synchronization failures resulting from such messages is sufficiently low. The standard requires that the UE should store in its memory the sequence numbers of a certain number of past successful authentication events. Such a mechanism ensures that a (stale) sequence number can still be accepted if it is among the last 32 sequence numbers generated (i.e. $\triangle = 32$ in Figure 2.3) and was not previously used. Unfortunately, the formal models of [33, 38] have ignored this important aspect of sequence number freshness verification, which renders their security and privacy analysis of 5G-AKA imprecise.

4.7 Summary and Recommendations

In this chapter we analyzed two recent attacks (AMA and LCA) on 5G subscription privacy by [38]. We established that AMA is infeasible in practice to execute in 5G networks. We also showed that LCA is trying to achieve what the existing LFM attack [31] already does with much less effort and greater effectiveness. Moreover, we demonstrated that both these attacks will become irrelevant if the LFM attack is patched. Additionally, we highlighted how the history of the symbolic modeling of the AKA protocol has been plagued with serious gaps that lead to various vulnerabilities. Looking at the results of our analysis in hindsight, it seems that the authors of [38] were overoptimistic in interpretation of their results. Keeping in mind the current development status of the 5G-AKA, the following recommendations are

made to 3GPP:

- To improve user privacy, 3GPP should consider appropriate countermeasures (as detailed in §3.5.3) for the LFM attack.
- Considering the aspects of protocol analysis discussed in §4.6.3, it is suggested that a comprehensive security and privacy analysis of the 5G-AKA in an appropriate *computational model* should be carried out.
- To prevent any further future attacks, there is a need to remediate the existing vulnerability of the 5G-AKA *identification phase* to replay attacks. We detail a proposal in Chapter 5 which prevents such replay attacks.

Chapter 5

An Alternative Proposal for SUPI Protection

In this chapter we present an alternative private identification scheme for 5G which utilizes only symmetric cryptographic primitives. We also provide a detailed formal security analysis of the scheme in a novel security framework. This scheme was published at the International Conference on Security Standardization Research 2018 [83].

5.1 Introduction

While many mobile users may be comfortable with the fact that their service provider is able to identify them and track their geographical location ubiquitously, fewer are likely to be comfortable with an arbitrary third party having this capability. In the hands of a third party, such a capability could lead to undesirable breaches of end-user privacy, opening the door to a range of potential consequences, such as harassment, stalking, employee monitoring, commercial profiling, etc. As elaborated earlier in §2.4, the subscribers are identified over the radio access link via frequently-changing temporary identifiers (called Temporary Mobile Subscriber Identity (TMSI) until 3G systems and a Globally Unique Temporary User Equipment Identity (GUTI) for 4G and 5G systems) by the serving network. However, despite the use of these temporary identifiers, IMSI-catching attacks [52, 53, 63, 104, 113, 118, 120] persist in today's mobile networks including the 4G LTE [108].

5.1.1 Countermeasures to IMSI-catchers in 5G

As highlighted in §3.3.1, IMSI-catching attacks have been a threat to all generations (2G/3G/4G) of mobile telecommunication [63]. As a result of technological barriers, this privacy problem appears to have persisted for decades [7]. However, 3GPP

decided to address this issue in 5G. In the event of identification failure via a 5G-GUTI, unlike earlier generations, 5G security specifications do not allow plaintext transmissions of the SUPI over the radio interface [14]. Instead, an *Elliptic Curve Integrated Encryption Scheme* (ECIES)-based privacy-preserving identifier containing the concealed SUPI is transmitted [131]. We elaborate upon the details of this scheme further in §5.4.1.

5.1.2 Motivation

5G Release 15 - the first full set of standalone 5G specifications - was finalized in June 2019. However, it will almost certainly take a decade or so before all legacy systems are upgraded to 5G. Hence, IMSI-catching attacks remain an issue during the mid-term future, possibly even beyond the year 2030. By then it is likely that practical quantum computers will pose a much more immediate threat than they do today [80, 45]. The impact of quantum computers on mobile networks is already being discussed within the telephony industry [105], with a call to implement quantum-secure cryptography [21]. It is thus imperative that 5G security specifications such as 3GPP TS 33.501 [14] (hereafter referred as TS 33.501) include options for quantum-secure schemes. Fortunately, 5G security has mostly relied upon symmetric cryptography (whose security is less impacted by quantum computers) for achieving its security objectives. However, the ECIES-based identification mechanism is an exception since it is known to be vulnerable to quantum algorithms. We suggest that one viable way forward is to develop a symmetric alternative to the ECIES mechanism. Any proposal for an alternative user identification protection scheme for 5G systems should ideally strive to satisfy the following requirements:

- Provision privacy guarantees such as *anonymity* and *unlinkability* [122] against a quantum adversary.
- The computational and communication overhead should be minimal. Specifically, the number of communication passes should not increase as it impacts the call-setup durations the most.
- Offer protection against replay attacks.
- Fulfill "Lawful Interception" requirements (for details see §2.9) in mobile telecommunications.

- Adhere to the existing 3GPP message structures as specified in current 5G specifications.

5.1.3 Chapter Contributions

The contributions of this chapter are as follows:

- We detail limitations of the ECIES-based identification scheme of TS 33.501.
- We present an alternative quantum-secure scheme which overcomes the limitations identified in the 3GPP scheme.
- We develop an appropriate model of security and formally prove the privacy guarantees offered by our proposal in this model.

The rest of this chapter is organized as follows: §5.2 discusses the related work while §5.3 details the pertinent aspects of the 5G-AKA. The current identity protection mechanism of 5G Release 15 is detailed in §5.4. §5.5 presents our identity protection proposal. §5.6 explains the security framework and §5.7 provides the analysis of our proposal. §5.8 provides a discussion about the impact of parameter sizes and §5.9 summarizes the chapter.

5.2 Related Work

To our knowledge, this is the first work on 5G identity protection since the publication of TS 33.501. Before a protection scheme was chosen, a study was conducted by 3GPP to evaluate a number of potential solutions. In total 24 proposals were considered, details of which can be found in the associated report 3GPP TR 33.899 (Clause 5.7.4) [8]. Most (but not all) proposals were based on public-key cryptography, and the ECIES-based mechanism was selected as the final candidate. The few symmetric-key proposals all relied on utilizing pseudonyms for privacy purposes, and thus were susceptible to desynchronization attacks potentially causing permanent DoS attacks on the mobile users.

Various academic works have considered IMSI-catching attacks. The major thrust of these papers has been to devise a solution for 3G/4G without modifying the existing message structures out of concern for legacy devices and backwards-compatibility.

Broek et al. [138] introduced a proposal based on changing pseudonyms and required no modifications to the existing infrastructure. As a result of reliance on changing pseudonyms, this solution was susceptible to desynchronization attacks. A similar proposal was that by Khan and Mitchell [87], which relied on using a set of IMSIs for a particular user to offer some degree of pseudonymity, however, as in the case of [138], this solution could also get knocked out of the service permanently. Khan and Mitchell, based upon their previous work, subsequently presented an improved solution [88]. This solution relied on using a dynamic pseudo-IMSI for identification purposes, however identity desynchronization attacks still had the potential to cause permanent denial of service. Thus their solution is accompanied with an identity recovery mechanism (in case of desynchronization) which required no changes to the existing message structures. However, this solution fails to satisfy the Lawful Interception (LI) requirements without further changes to the existing message structures.

5.3 The 5G-AKA

Our proposal for private identification in 5G (§5.4) works in tandem with the 5G-AKA protocol. Hence, before we layout our proposal, we detail the message flow of the 5G-AKA. As already elaborated earlier in §2.7, 5G-AKA utilizes various symmetric cryptographic algorithms. Detail of how these cryptographic algorithms are used for calculation of various 5G-AKA parameters can be found in Table 5.1. A pictorial representation of the 5G-AKA message flow is given in Figure 5.1 and elaborated further in the following:

0. [1] To initiate authentication, the UE sends the SN either the 5G-GUTI in a "registration request" message or the SUCI as response to an "identifier request" message (see §2.6 for further details).

1. In case of a 5G-GUTI, the SN extracts the corresponding SUPI from its database and forwards it along with its serving network name (SN_{name}) to the HN in an "authenticate request" message. Otherwise the SUCI is sent instead of the SUPI.

2. If the SUCI is received in an authenticate request message by HN, it de-conceals (for details see §5.4.1) the SUPI from it. It further derives the expected re-

[1] This first Step is numbered 0 because its not an exclusive part of the AKA but rather the identification phase.

Table 5.1: Description of 5G-AKA parameters.

Parameter	**Content/Description**
$RAND$	128-bit Random Challenge
SQN	48-bit Sequence Number
AMF	16-bit Authentication Management Field
SN_{name}	Serving Network Name
AK	$f_5(K, RAND)$
CK	$f_3(K, RAND)$
IK	$f_4(K, RAND)$
RES	$f_2(K, RAND)$
MAC	$f_1(K, SQN \| RAND \| AMF)$
$AUTN$	$(SQN \oplus AK) \| AMF \| MAC$
$RES^*/XRES^*$	$KDF(CK \| IK, SN_{name} \| RAND \| RES/XRES)$
$HXRES^*/HRES^*$	$SHA256(RAND \| XRES^*/RES^*)$
K_{AUSF}	$KDF(CK \| IK, SN_{name} \| SQN \oplus AK)$
K_{SEAF}	$KDF(K_{AUSF}, SN_{name})$
$5G\ AV$	$RAND \| AUTN \| HXRES^*$

sponse XRES* and generates the authentication vector 5G AV. The 5G AV consists of a random challenge RAND, an authentication token AUTN and a hash of expected response HXRES*.

3. The HN stores XRES*.

4. The HN forwards the 5G AV (RAND, AUTN, HXRES*) in an "authenticate response" message to the SN.

5. The SN forwards RAND, AUTN to the UE in an Auth-Req message.

6. Upon receiving the RAND and AUTN, the UE verifies the freshness and authenticity as described in [9]. It then computes the response RES* and derives the anchor key K_{SEAF} to be used for establishment of the secure channel with the SN.

7. The UE returns RES* in an Auth-Resp message to the SN.

8. The SN then computes the hash of the response HRES* from the received RES* and compares HRES* with XHRES*. If they are equal, the SN considers the authentication successful.

9. The SN then sends RES*, as received from the UE, to the HN in an "authentication confirmation" message (containing the SUPI or SUCI and the serving network name).

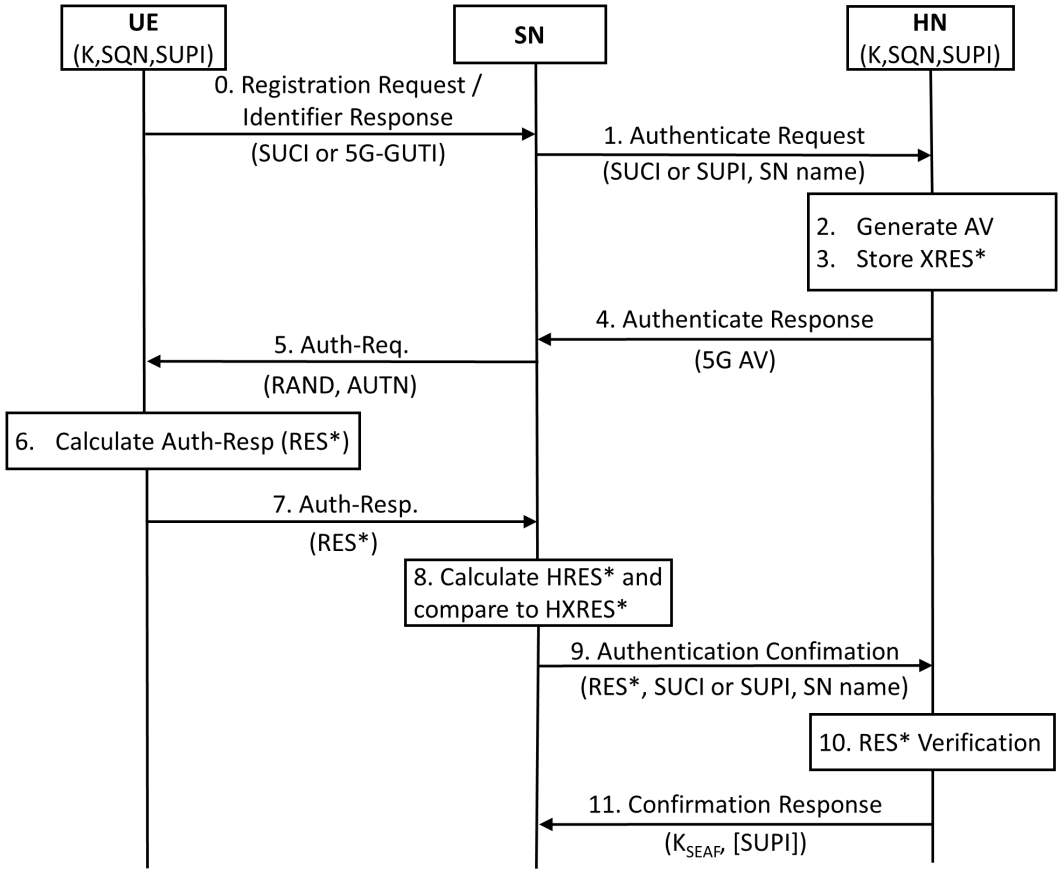


Figure 5.1: Overview of the 5G-AKA protocol.

10. When the HN receives a confirmation message, it compares RES* with the stored XRES*. If these two are equal, the HN considers the confirmation message as successfully verified.

11. Finally, the HN indicates to the SN in a "confirmation response" message whether the confirmation was successful or not. In case of a success, an anchor key K_{SEAF} which is cryptographically bound to the requesting SN is also provided by the HN. If the HN received a SUCI from the SN when authentication was initiated, and if the confirmation is successful, then the HN also includes the SUPI in this message.

5.4 Identity Privacy in 5G

In the 5G system, Subscription Concealed Identifier (SUCI) is a privacy preserving identifier containing the concealed SUPI. The UE generates a SUCI using a protection scheme (see §5.4.1) with the public key of the HN that was securely provisioned

to the USIM during the USIM registration. Only the MSIN part of the SUPI is concealed by the protection scheme, while the home network identifier (MCC/MNC) is transmitted in plaintext. The data fields constituting the SUCI are:

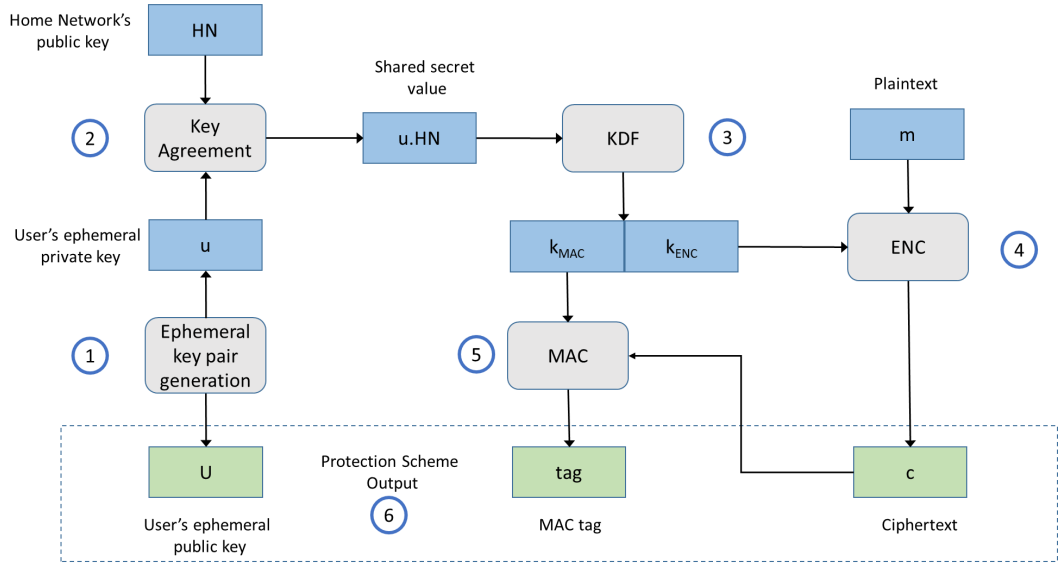
- **Protection Scheme Identifier.** This field represents the null scheme[2] or any other specified protection scheme.
- **Home Network Public-Key Identifier.** This represents the public key provisioned by the HN. In case of a null scheme, this field is set to null.
- **Home Network Identifier.** This contains the MCC and MNC part of the SUPI.
- **Protection Scheme Output.** This represents the output of the public-key based protection scheme.

The subscriber identification mechanism of 5G allows the identification of a UE on the radio path by means of the SUCI. This mechanism is usually invoked by the SN by sending an *identifier request* message (§2.6) to the UE when the UE is not identifiable by means of a temporary identity. The UE then responds with the *identifier response* message (§2.6), containing the SUCI. Additionally, if the UE sends a *registration request* message (§2.6) of the type "initial registration" to a mobile network for which it does not already have a 5G-GUTI, then the UE includes a SUCI to the *registration request*.

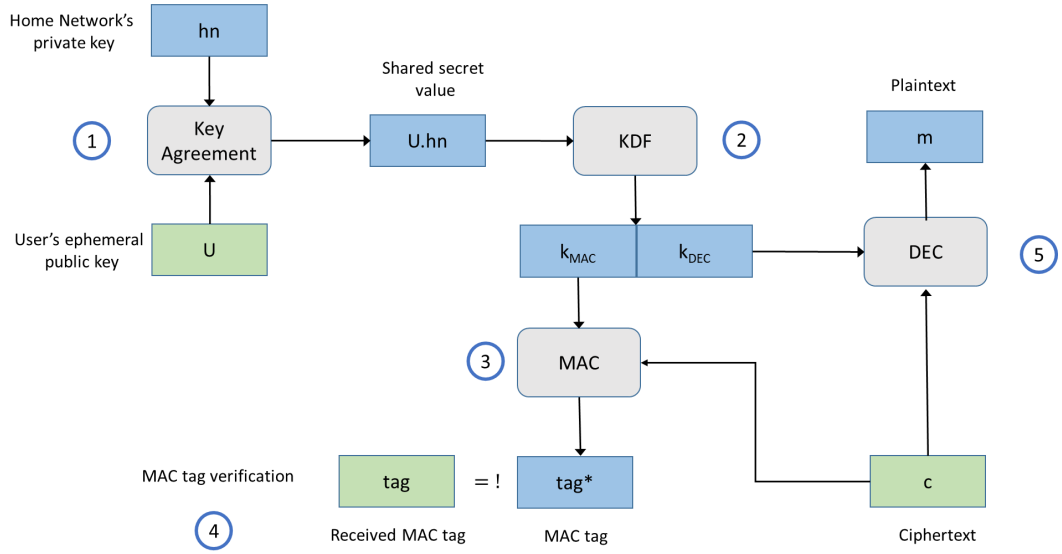
5.4.1 ECIES-based Protection Scheme

We now provide an overview of the ECIES-based protection scheme as described in TS 33.501 (Annex C.3) [14]. ECIES [131] is a hybrid encryption scheme that combines Elliptic Curve Cryptography (ECC) [69] with symmetric-key cryptography; it is a semantically secure probabilistic encryption scheme ensuring that successive encryptions of the same plaintext with the same public key result in different ciphertexts with very high probability. To compute a fresh SUCI, the UE generates a fresh ECC ephemeral public/private key pair utilizing the HN public key. Processing on the UE side is done according to the encryption operation defined in [128] and as further illustrated in Figure 5.2a. The final output of this protection scheme is

[2]The null-scheme is used only if the UE is making an unauthenticated emergency session or if the HN has configured "null-scheme" to be used or if the HN has not provisioned the public key needed to generate SUCI.



(a) Encryption at UE side



(b) Decryption at HN side

Figure 5.2: Detail of ECIES-based protection scheme

the concatenation of the ECC ephemeral public key, the ciphertext value, the MAC tag value, and any other parameters, if applicable. The HN uses the received ECC ephemeral public key and its private key to deconceal the received SUCI. Processing on the HN side is illustrated in Figure 5.2b.

The ECIES-based protection scheme is a framework, not a concrete algorithm. It can be implemented by plugging different algorithms, e.g. the secp256k1 or P-521 elliptic curve for the public-key calculations, PBKDF2 or Scrypt for KDF func-

tion, AES-CTR or AES-GCM or ChaCha20-Poly1305 for symmetric cipher and authentication tag, HMAC-SHA512 for MAC algorithm, etc. Hence, TS 33.501 includes two ECIES profiles which specifies the exact algorithms to be used for various cryptographic operations within the ECIES framework, both for the approximately 128-bit security level. Both profiles use AES-128 in CTR mode for confidentiality, ANSI-X9.63-KDF for KDF and HMAC-SHA-256 for authenticity in the symmetric-key cryptography part, but use either Curve25519 or secp256r1 elliptic curves for the public-key cryptography part.

5.4.2 Limitations of the 3GPP Protection Scheme

Although the ECIES-based scheme is oblivious to loss of synchronization between the UE and HN and has simple key management requirements, both of which lead to significant reduction in connection failures, there are still aspects which require further improvement [59].

- **Quantum Insecurity.** As the ECIES-based scheme employs ECC to provision identity privacy, it relies on the hardness assumption of the Elliptic Curve Discrete Logarithm Problem (ECDLP) [144]. An adversary capable of issuing quantum queries to an appropriate quantum computer can easily break this scheme by employing Shor's algorithm [130].
- **Chosen SUPI Attacks.** Any arbitrary third party can always select a SUPI of its choosing and send the corresponding SUCI to the HN. Thereafter the adversary can look out for various responses from the HN, depending on whether the target user is present in that particular cell tower area or not. Any noticeable variation in the perceived output would allow the adversary to confirm or deny the presence of the target in that particular cell. There is no mechanism in the ECIES-based scheme to prevent these attacks.
- **Replay Attacks.** Note that the ECIES-based scheme does not have any inherent mechanism to provide freshness guarantees to the HN and is thus susceptible to replay attacks. An adversary can always resend a previously encrypted SUPI to the HN and look out for various responses (such as authentication challenge or a failure message). Based on the received response, a device whose SUPI is unknown to the attacker may be tracked with some confidence.

- **Downgrade Attacks.** An active adversary simulating a (false) base station can force the UE to use one of the previous generation (GSM/UMTS/LTE) and can then get hold of the IMSI / SUPI using an *identity request* message. In 3GPP Release 15 [14], the SUPI is derived directly from the IMSI, so these downgrade attacks also compromise the 5G SUPI.
- **Update of HN Public Key.** There could be situations which require the HN to have a robust way of quickly updating its public key to subscriber UEs, such as a malware attack which tries to recover the home network's private key. Such situations enforce the need to have a quick way of updating the corresponding public keys.

5.5 Towards Quantum-secure Identity Privacy

We now detail our proposal for an alternative identity protection scheme. Unlike the ECIES-based scheme (§5.4.1), our proposal mostly requires the cryptographic primitives already provisioned by the current 5G specifications. We utilise the previously specified key derivation and message authentication functions of the 5G-AKA for our proposal. Specifically, we use function f_1 for message authentication and functions f_3, f_4, f_5 and $f_5{}^*$ for key derivation. As elaborated in 3GPP TS 33.102, no valuable information can be inferred from the values of any of these functions about other functions [9]. Table 5.2 gives a summary of notations used in the proposed scheme and Figure 5.3 provides an overview of the proposed scheme PQID. Various phases of the PQID are explained further.

5.5.1 System Setup Phase

The HN generates a long-term secret key K_{HN} for the calculation of identification parameters for its subscribers. HN stores this value internally in a secure manner, allowing no other entity access. HN randomly chooses K_N and K_{ID} during the USIM registration and computes the (data) confidentiality key $CKID = f_4(K_{HN}, K_N)$ for the protection scheme as well as identification parameters $A = SUPI \oplus CKID$ and $B = K_{HN} \oplus K_N$. In addition to the $SUPI$, the AKA sequence number SQN and the shared key K (which are all from the original 5G-AKA), the USIM also stores K_{ID}, identification parameters A and B along with an additional 48-bit identifica-

Table 5.2: Notation used in the proposed scheme.

Notation	Description
A and B	Identification parameters generated by HN for UE
$SQNID$	Counter used for replay prevention
K_{HN}	Long term secret key of HN
K_N	Randomly generated ephemeral parameter
K_{ID}	Randomly generated long-term secret key for UE
$RANDID$	Freshly generated random number
$CKID$	Confidentiality key
$AKID$	Anonymity key
$MACID$	MAC Tag
f_1	Message Authentication Function
f_3, f_4, f_5, $f_5{}^*$	Key Derivation Functions
AE.Enc	Authenticated Encryption Function
AE.Dec	Authenticated Decryption Function
$f(K, X)$	Execution of keyed-function f upon input X with key K

tion sequence number $SQNID_{UE}$[3] with initial value set to 1. HN initializes a corresponding identification sequence number $SQNID_{HN}$[4] with initial value of 0 and stores $SQNID_{HN}$ in its database. HN also stores the value of K_{ID} in its database for the particular subscriber. An algorithmic description of the computation of this phase can be found in Figure 5.4.

5.5.2 Identification Phase

An algorithmic description of the operations of UE and HN during this phase is presented in Figure 5.5. Note that the output of $f_3(K_{ID}, RANDID)$ is truncated to get a 48-bit $AKID$. The UE prepares the $SUCI = (\texttt{label}_{\texttt{ps}}, \epsilon, \texttt{label}_{HN}, (D\|A\|B\|C\|MACID))$ using various data fields[5], as explained in §5.4, and forwards $SUCI$ to SN. The SN appends its SN_{name} (Clause 6.1.1.4 of [14]) to the received SUCI and forwards the resulting message to the HN. Upon successful MAC verification, HN accepts the extracted $SUPI$ as valid for subsequent processing.

[3] As the ME and the USIM together form the UE and the trust model within the UE is reasonably simple i.e. there are two trust domains, the tamper proof UICC on which the the USIM resides as trust anchor and the ME; for sake of simplicity, we label user side notations with UE instead of distinct USIM or ME.

[4] Note that HN will maintain a separate distinct value of $SQNID_{HN}$ for each registered USIM in its database.

[5] Note that $\texttt{label}_{\texttt{ps}}$ is a constant value indicating the protection scheme, and $\texttt{label}_{HN}$ is a constant value identifying the HN.

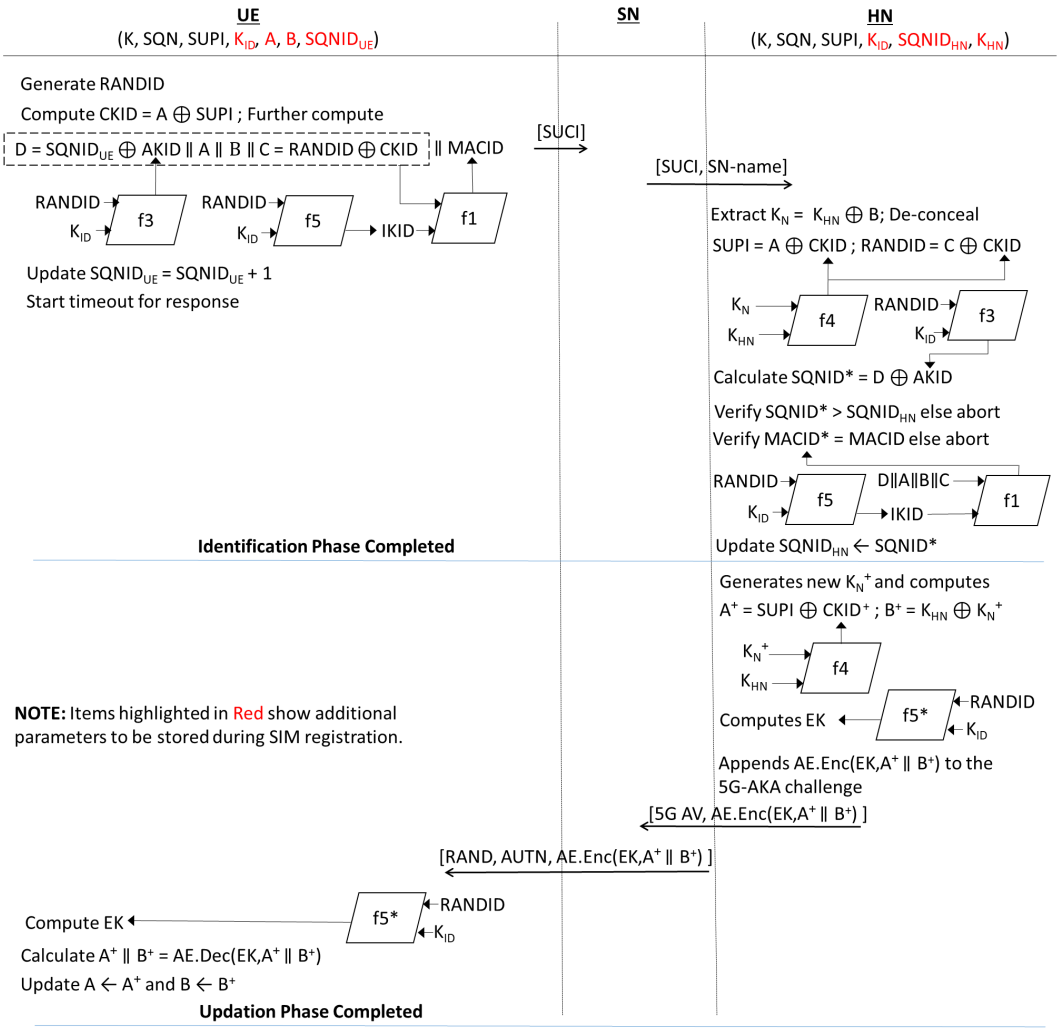


Figure 5.3: Our proposed protection scheme PQID.

$K_{HN} \xleftarrow{\$} \{0,1\}^{\lambda}$ // init master secret key K_{HN}
for *each USIM* **do**
  // init secret keys and identification parameters
  $K_N \xleftarrow{\$} \{0,1\}^{\lambda}$
  $K_{ID} \xleftarrow{\$} \{0,1\}^{\lambda}$
  $SQNID_{UE} \leftarrow 1$
  $SQNID_{HN} \leftarrow 0$
  $CKID \leftarrow f_4(K_{HN}, K_N)$
  $A \leftarrow SUPI \oplus CKID$
  $B \leftarrow K_{HN} \oplus K_N$
end
$USIM \leftarrow (K_{ID}, A, B, SQNID_{UE})$
$HN \leftarrow (K_{ID}, SQNID_{HN})$

Figure 5.4: Algorithmic description of *system setup phase*.

```
// description of UE's operations
RANDID ←$ {0,1}^λ
CKID ← A ⊕ SUPI
AKID ← f3(K_ID, RANDID)
IKID ← f5(K_ID, RANDID)
C ← RANDID ⊕ CKID
D ← SQNID_UE ⊕ AKID
MACID ← f1(IKID, D‖A‖B‖C)
SUCI ← (label_ps, ϵ, label_HN, D‖A‖B‖C‖MACID)
SQNID_UE ← SQNID_UE + 1

// description of HN's operations
K_N ← K_HN ⊕ B
CKID ← f4(K_HN, K_N)
SUPI ← A ⊕ CKID
RANDID ← C ⊕ CKID
AKID ← f3(K_ID, RANDID)
IKID ← f5(K_ID, RANDID)
SQNID* ← D ⊕ AKID
if SQNID* ≤ SQNID_HN then
    abort
else
    MACID* ← f1(IKID, D‖A‖B‖C)
end
if MACID ≠ MACID* then
    abort
else
    SQNID_HN ← SQNID*
end
```

Figure 5.5: Algorithmic description of *identification phase*.

5.5.3 Update Phase

An algorithmic description of the operations of UE and HN during this phase can be found in Figure 5.6. The output of the encryption scheme $\mathsf{AE.Enc}(EK, A^+\|B^+)$ is appended to the 5G-AKA authentication vector 5G AV and is forwarded to the SN as part of the *authenticate response* message (Step 4 in Figure 5.1) of the 5G-AKA. The SN, upon receipt of the response message, undertakes the required steps necessary for 5G-AKA and forwards the encrypted identification parameters to the UE along with the 5G-AKA authentication challenge parameters $RAND$ (note that $RAND$ is unrelated to $RANDID$) and $AUTN$ (Step 5 in Figure 5.1).

// description of HN's operations
$K_N{}^+ \leftarrow \{0,1\}^\lambda$
$CKID^+ \leftarrow f_4(K_{HN}, K_N{}^+)$
$A^+ \leftarrow SUPI \oplus CKID^+$
$B^+ \leftarrow K_{HN} \oplus K_N{}^+$
$EK \leftarrow f_5{}^*(K_{ID}, RANDID)$
$EKID \leftarrow \mathsf{AE.Enc}(EK, A^+ \| B^+)$

// description of UE's operations
$EK \leftarrow f_5{}^*(K_{ID}, RANDID)$
$A^+ \| B^+ \leftarrow \mathsf{AE.Dec}(EK, EKID)$
$A,\ B \leftarrow A^+,\ B^+$

Figure 5.6: Algorithmic description of *update phase*.

5.6 Security Framework

In this section we introduce our Symmetric Updatable Private Authentication (SUPA) framework, which follows in the long tradition of standard Bellare-Rogaway (BR) key-indistinguishability games. Essentially, a protocol within the SUPA framework is a protocol that authenticates an end-user to a central node via a shared symmetric key in a private way. In comparison to similar BR-styled mutual authentication games, our SUPA experiment diverges by considering *identity privacy*. In particular, the SUPA-based security experiment asks the adversary to decide which of two parties attempted to authenticate itself to a centralised home network. In addition, SUPA distinguishes itself by considering a *multi-stage* authentication protocol - i.e. subsequent authentication attempts between the UE and the HN (after the first successful authentication) are not independent, but instead dependent on values derived from previous stages. This allows us to capture both *identity confidentiality* and *untraceability* from the 3GPP requirements of user privacy (see §5.1). We can now formally define a SUPA protocol.

Definition 1 (Symmetric Updatable Private Authentication)**.** *A Symmetric Updatable Private Authentication (SUPA) protocol is a tuple of algorithms* $\{\mathsf{SetupHN}, \mathsf{SetupUE}, \mathsf{Identify}, \mathsf{Update}\}$.

- $\mathsf{SetupHN}(\lambda) \rightarrow K_{HN}$*:* SetupHN *takes as input some security parameter* λ *and outputs a long-term symmetric key* K_{HN}.
- $\mathsf{SetupUE}(\lambda, K_{HN}) \rightarrow K, st$*:* SetupUE *takes as input some security parameter* λ

and a long-term symmetric key K_{HN}, and outputs some shared (between the UE and the HN) secret state st and a shared symmetric key K.

- $\mathsf{Identify}(role, m, st, K_{HN}) \rightarrow (id, m', st')$*:* Identify *takes as input the role of the party in the protocol execution, a (potentially empty) message m, the internal state of the party st and (if role = HN) the long-term HN key K_{HN}, and outputs an identifier id, a new (potentially empty) message m', and an updated state st'. Note that the identifier id doubles as a failure flag if the* Identify *algorithm is forced to abort.*

- $\mathsf{Update}(role, m, st, K_{HN}) \rightarrow (m', st')$*:* Update *takes as input the role of the party in the protocol execution, a (potentially empty) message m, the internal state of the party st and (if role = HN) the long-term HN key K_{HN}, and outputs a new (potentially empty) message m', an updated state st'. As in* Identify*, the output message m' doubles as a failure flag if the* Update *algorithm is forced to abort.*

5.6.1 Execution Environment

We now describe the execution environment of the SUPA security experiment. The experiment $\mathsf{Exp}^{\mathsf{SUPA}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$ is played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The challenger $\mathcal{C}$ maintains a single HN, running a number of instances of the SUPA protocol Π, and a set of (up to) n_N users $UE_1, \ldots, UE_{n_N}$ (representing nodes communicating with the home network HN), each potentially running a single session executing (up to) n_S consecutive stages of Π. The protocol Π is represented as a tuple of algorithms $\mathsf{SUPA} = \{\mathsf{SetupHN}, \mathsf{SetupUE}, \mathsf{Identify}, \mathsf{Update}\}$. We abuse notation and use π_i^s to refer to both the identifier of the s-th stage of Π being run by node UE_i and the collection of per-session variables maintained for this stage. Each session maintains the following set of per-session variables:

- $i \in \{1, \ldots, n_N\}$ - the index of the party UE_i;
- $ltk \in \{0,1\}^\lambda$ - the long-term symmetric secret of UE_i, shared with HN;
- $id \in \{0,1\}^*$ - the identifier of party UE_i;
- $m_s \in \{0,1\}^* \cup \{\perp\}$ - the concatenation of messages sent by the session, initialised by $\perp$;

- $m_r \in \{0,1\}^* \cup \{\perp\}$ - the concatenation of messages received by the session, initialised by $\perp$;
- $st \in \{0,1\}^* \cup \{\perp\}$ - the per-stage secret state of the session, initialised by $\perp$;
- $s \in \{1,\ldots,n_S\}$ - the index of the most recently completed authentication stage, initialised by 1 and increased monotonically;
- $\alpha \in \{\texttt{active}, \texttt{accept}, \perp\}$ - the current status of the session, initialised by $\perp$.

Our experiment begins with the challenger $\mathcal{C}$ sampling the random test bit $b \xleftarrow{\$} \{0,1\}$. The challenger generates the long-term symmetric key K_{HN} of the HN and initializes its corruption registers (which maintain the list of secrets $\mathcal{A}$ has leaked). At this point, $\mathcal{A}$ now gains access to the queries listed in §5.6.2 and eventually terminates and outputs a single guess bit b'. The freshness predicate fresh for our SUPA security experiment is defined next.

Definition 2 (SUPA-fresh)**.** *A session π_i^s in the* SUPA *experiment is* fresh *if* $\mathsf{clean}(\pi_i^s) = \texttt{true}$ *(as defined in Definition 3) and $\pi_i^s.m_r = \perp \wedge \pi_i^s.m_s = \perp$ at the start of the experiment.*

If $\mathcal{A}$ causes the challenger to:

- either execute $\mathsf{Identify}(HN, m, HN.st, K_{HN}) \rightarrow (id, m', st')$ such that there exists some session $\pi_i^s.id = id$, but $m \not\subset \pi_i^s.m_s$[6] or;
- execute $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon) \rightarrow (m', st')$ such that $m' \neq \perp$ but there was no execution of $\mathsf{Update}(HN, m^*, HN.st, K_{HN}) \rightarrow (m, HN.st')$.

If either of the above is true and $\mathsf{fresh}(i,s) = \texttt{true}$ then $\mathcal{C}$ returns 1. Otherwise, if $\mathcal{A}$ issued a $\mathbf{Test}(i^*, s^*)$ query, then $\mathcal{C}$ computes $\mathsf{fresh}(i^*, s^*)$. If $\mathsf{fresh}(i^*, s^*)$ is $\texttt{true}$, then the challenger returns $(b = b')$, otherwise the challenger returns $b^* \xleftarrow{\$} \{0,1\}$.

5.6.2 Adversary Queries

Here we describe the intuition behind each query that $\mathcal{A}$ has access to during the SUPA experiment. For full details on each of these queries, see Figure 5.7.

[6] Note that here we are using $\subset$ to indicate substrings

$\mathsf{Exp}^{\mathsf{SUPA},\mathsf{clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$:

$b \xleftarrow{\$} \{0,1\}$
$K_{HN} \xleftarrow{\$} \mathsf{SetupHN}(\lambda)$
$\mathsf{LSKflag}_i, \ldots, \mathsf{LSKflag}_{n_N} \leftarrow \texttt{clean}$
$\mathsf{PSSflag}^1_1, \ldots, \mathsf{PSSflag}^{n_N}_{n_S} \leftarrow \texttt{clean}$
$ctr \leftarrow 0$
$b' \xleftarrow{\$} \mathcal{A}^{\mathsf{Send}\star,\mathsf{Create},\mathsf{Corrupt},\mathsf{StateReveal}}(\lambda)$
if $\exists\ (i^*, s^*)$ s.t. $((\mathsf{Identify}(HN, m, HN.st, K_{HN}) \rightarrow (id, m', st)$ s.t $\pi^{s^*}_{i^*}.id = id, m \neq \pi^{s^*}_{i^*}.m_s) \wedge (\mathsf{clean}(\pi^{s^*}_{i^*}))) \vee ((\mathsf{Update}(UE, m, \pi^{s^*}_{i^*}.st, \epsilon) \rightarrow (m', st')$ s.t. $m' \neq \perp, \nexists \mathsf{Update}(HN, m^*, HN.st, K_{HN}) \rightarrow (m, HN.st'))) \wedge (\mathsf{clean}(\pi^{s^*}_{i^*}))$ **then**
  return 1
end if
if $\mathsf{clean}(\pi_b) \wedge \mathsf{clean}(\pi_{1-b})$ **then**
  return $(b' = b)$
else
  return $b^* \xleftarrow{\$} \{0,1\}$
end if

Test$((i,s),(i',s'))$:

if $(\pi^s_i.\alpha = \texttt{active}) \vee (\pi^{s'}_{i'}.\alpha = \texttt{active})$ **then**
  return $\perp$
end if
if $(b = 0)$ **then**
  $\pi_b \leftarrow \pi^s_i$
  $\pi_{b-1} \leftarrow \pi^{s'}_{i'}$
else
  $\pi_b \leftarrow \pi^{s'}_{i'}$
  $\pi_{b-1} \leftarrow \pi^s_i$
end if
$m \leftarrow \mathsf{Identify}(UE, \perp, \pi_b.st, \perp)$
return m

StateReveal(i,s):

if $\pi^s_i.st = \perp$ **then**
  return $\perp$
end if
$\mathsf{PSSflag}^i_s \leftarrow \texttt{corrupt}$
return $\pi^s_i.st$

Create(λ):

$ctr \leftarrow ctr + 1$
$\pi.s \leftarrow 1$
$\pi.ltk, \pi.st \leftarrow \mathsf{SetupUE}(\lambda, K_{HN})$
$\pi.i \leftarrow ctr$
return $\pi.i$

SendTest(m):

$\mathsf{Send}(\pi_b, m) \rightarrow m'$
return m'

Send$(role, i, s, m)$:

if $role = HN$ **then**
  $(HN.st', m') \leftarrow F(\lambda, HN, m)$
end if
let $s = max\{s : \pi^s_i.\alpha \neq \perp\}$
if $\pi^s_i.\alpha \neq \texttt{active}$ **then**
  return $\perp$
end if
$\pi^s_i.m_r \leftarrow \pi^s_i.m_r \| m$
$(\pi^s_i, m') \leftarrow F(\lambda, \pi^s_i, m)$
$\pi^s_i.m_s \leftarrow \pi^s_i.m_s \| m'$
return m'

Corrupt(i):

$\mathsf{LSKflag}_i \leftarrow \texttt{corrupt}$
return $\pi_i.ltk$

Figure 5.7: An algorithmic description of the SUPA security experiment. We assume the existence of a function F that is capable of taking as input a message m and the current internal state $\pi^s_i.st$ of the protocol execution and forwarding the inputs to either Update or Identify as appropriate. We refer to the "test" session in the description of the SUPA experiment as π_b (and the other session as π_{1-b}).

- **Create**(i): Allows $\mathcal{A}$ to initialise a new UE party with shared symmetric state and shared symmetric key with HN.
- **Send**$(role, i, s, m) \rightarrow m'$: Sends a message m to session π_i^s, which updates the per-session variables, returning a (potentially empty) message m'.
- **Corrupt**$(i) \rightarrow \pi_i.ltk$: Reveals to $\mathcal{A}$ the long-term symmetric key of UE_i[7].
- **Test**$(i, s, i', s') \rightarrow m$: Uses the random bit b sampled by $\mathcal{C}$ to begin a new $\mathsf{Identify}$ phase with either π_i^s (if $b = 0$) or $\pi_{i'}^{s'}$ (if $b = 1$). For ease of notation, we refer to the "test" session as π_b (and the other session as π_{1-b}). Note that $\mathcal{A}$ cannot issue this query if there exists some stage s such that either $\pi_i^s.\alpha = \texttt{active}$ or $\pi_{i'}^{s'}.\alpha = \texttt{active}$, nor can $\mathcal{A}$ issue **Send** queries to π_i^s or $\pi_{i'}^{s'}$ until π_b has either accepted or rejected the protocol execution.
- **SendTest**$(m) \rightarrow m'$: Allows $\mathcal{A}$ to send a message m to the test session π_b after $\mathcal{A}$ has issued a **Test** query. After $\pi_b.\alpha \neq \texttt{active}$, the challenger responds to **SendTest** queries with $\perp$.
- **StateReveal**$(i, s) \rightarrow \pi_i^s$: Reveals to $\mathcal{A}$ the full internal state of π_i^s.

We require a cleanness predicate, in order to disallow combinations of **Corrupt** and **StateReveal** queries that allow an adversary to trivially break $SUPA$ security. We do not capture notions of forward secrecy, so our cleanness predicate is very simple: $\mathcal{A}$ is not allowed to break sessions that it has issued either a **Corrupt** or a **StateReveal** query to.

Definition 3 (SUPA-clean)**.** *A session π_i^s in the* SUPA *experiment defined in Figure 5.7 is* clean *if* $\mathsf{LSKflag}_i \neq \texttt{corrupt}$ *and* $\mathsf{PSSflag}_i^s \neq \texttt{corrupt}$.

5.6.3 Security Definitions

Here we define the security of a SUPA protocol, and additionally show that the PQID scheme described in Figure 5.3 executes correctly in the presence of a passive adversary.

Definition 4 (Private Authentication Security)**.** *Let* Π *be a* SUPA *protocol, and* n_N, $n_S \in \mathbb{N}$. *For a given cleanness predicate* clean, *and a PPT algorithm* $\mathcal{A}$, *we define the advantage of* $\mathcal{A}$ *in the* SUPA *game to be:*

$$\mathsf{Adv}^{\mathsf{SUPA,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{SUPA,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|.$$

[7] In PQID, this is the key K_{ID}, not the independent 5G-AKA key K.

We say that Π *is* SUPA*-secure if, for all* $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{SUPA,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible in the security parameter* λ.

We also need to define *identification correctness* as well as *update correctness*, to ensure that we only capture protocols that are actually useful.

Definition 5 (Identification Correctness)**.** *Let* Π *be a* SUPA *protocol. We say that* Π *has* identification correctness *if after an execution of* $\mathsf{Identify}(HN, m', HN.st, K_{HN}) \rightarrow (id', m^*, st')$ *in the presence of a passive adversary* $\mathcal{A}$ *such that for some session* $\pi_i^s.m_s = m'$*, then* $\pi_i^s.id = id'$.

It is fairly straightforward to see that the proposed protocol in Figure 5.3 has identification correctness: The fields $A = SUPI \oplus CKID$ and $B = K_{HN} \oplus K_N$ sent by the UE contains all the information necessary to recompute the identifier $SUPI$ of the UE. HN first computes $K_N = B \oplus K_{HN}$ and then $CKID = f_4(K_{HN}, K_N)$. Retrieving $SUPI$ is then simply a matter of $SUPI \leftarrow A \oplus CKID$.

Update correctness is a little different to identification correctness. We only require that the session executing an Update using output from HN simply updates their state without aborting the protocol execution, instead of having to agree to some shared updated state. This is to capture stateless HN sessions that simply regenerate per-session state when required, usually by processing client-maintained tokens. In this sense, the A and B values sent by the UE during our PQID protocol are tokens that allow HN to recover per-session state.

Definition 6 (Update Correctness)**.** *Let* Π *be a* SUPA *protocol. We say that* Π *has* update correctness *if after an execution of* $\mathsf{Update}(UE, m', \pi_i^s.st, \epsilon) \rightarrow (m^*, \pi_i^s.st')$ *in the presence of a passive adversary* $\mathcal{A}$ *such that for some execution of* $\mathsf{Update}(HN, m, HN.st, K_{HN}) \rightarrow (m', HN.st')$*, then* $m^* \neq \perp$ *and* $\pi_i^s.st' \neq \pi_i^s.st$.

Similarly to identification correctness, it is straightforward to see that the proposed protocol in Figure 5.3 has update correctness: The fields $A^+ = SUPI \oplus CKID^+$ and $B^+ = K_{HN} \oplus {K_N}^+$ encrypted under $EK = f_{5^*}(K_{ID}, RANDID)$ sent by the HN contains all the information necessary to update the values A, B and $CKID$. UE computes $EK = f_{5^*}(K_{ID}, RANDID)$ (where $RANDID$ was sampled initially by UE and K_{ID} is the long-term symmetric key shared by UE and HN in the PQID protocol, so both are known to UE), and decrypts A^+ and B^+. Afterwards, UE updates $A \leftarrow A^+$, $B \leftarrow B^+$, $CKID \leftarrow A^+ \oplus SUPI$.

5.7 Analysis of the Proposed Protection Scheme

5.7.1 Formal Analysis

We discuss and analyse our proposed 5G identity protection scheme PQID within the SUPA security framework, and show that it achieves the notion of SUPA protocols.

Theorem 1. *The identity protection scheme* PQID *given in Figure 5.3 is* SUPA*-secure under cleanness predicate* clean *and assuming all hash functions are random oracles. For any PPT algorithm* $\mathcal{A}$ *against the* SUPA *experiment,* $\mathsf{Adv}^{\mathsf{SUPA,clean}}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible under the* ae, kdf *and* eufcma *security assumptions of the* AE, KDF *and* MAC *schemes, respectively.*

Proof. Before we begin our analysis in earnest, we show that an adversary $\mathcal{A}$ is unable to recover the long-term symmetric key of the home network K_{HN} (with non-negligible probability) even if $\mathcal{A}$ reveals all long-term secrets K of all nodes and all per-stage secret states st, assuming underlying hash functions are random oracles. In our proof we work within the random oracle model, and $\mathcal{A}$ cannot learn anything about K_{HN} from hash outputs $\mathsf{H}(K_{HN}, X)$ (where X is any concatenation of arbitrary values).

We turn to $\mathcal{A}$ attempting to learn K_{HN} that has been "blinded" through exclusive-or (XOR) operations, which are only sent in the following values: $B = K_{HN} \oplus K_N$ and $B^+ = K_{HN} \oplus {K_N}^+$. K_N and ${K_N}^+$ are acting as one-time-pads encrypting the long-term symmetric key of the home network HN, and each $K_N/{K_N}^+$ is a value internal to the home network that cannot be compromised via $\mathcal{A}$ issuing a **Corrupt** or **StateReveal** query. $\mathcal{A}$ therefore cannot recover K_{HN} in this way, but can attempt to guess and verify the guess by first querying **StateReveal** to any UE party, recovering $CKID$ and B, and querying the random oracle with $({K_{HN}}', B \oplus {K_{HN}}')$ and comparing the output of the random oracle with $CKID$. The probability of $\mathcal{A}$'s success in this strategy is $q_r/2^{\lambda-1}$. (where q_r is the number of queries that $\mathcal{A}$ makes to the random oracle and λ is the bit-length of K_{HN}). During our analysis then, we assume that in each stage of a protocol execution K_{HN} is indistinguishable from a uniformly-random value ${K_{HN}}^*$ from the same distribution.

In our analysis, we split our proof into three cases:

1. $\mathcal{A}$ has caused a session π_i^s to reach a status `accept` when calling $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon)$ such that m is not the output of HN and $\mathsf{clean}(\pi_i^s) = \texttt{true}$.

2. $\mathcal{A}$ has caused HN to call $\mathsf{Identify}(HN, m, HN.st, K_{HN}) \rightarrow (id', m', HN.st')$ such that $\exists\ \pi_i^s.id = id'$, but m was not the output of $\mathsf{Identify}(UE, \epsilon, \pi_i^s.st, \epsilon)$ and $\mathsf{clean}(\pi_i^s) = \texttt{true}$.

3. $\mathcal{A}$ has output a guessed bit b' after issuing a $\mathbf{Test}(i, s, i', s')$ query

We show that $\mathcal{A}$ has negligible advantage in causing the first two cases to occur, and thus $\mathcal{A}$ also has negligible advantage in winning the SUPA experiment in the third case. Each of these three cases are disjoint by the definition of the SUPA experiment: the experiment terminates immediately in the first two cases when a (clean) session has reached the `accept` state after recieving a message m that is not the honest output of either the HN or the UE. It follows that $\mathcal{A}$ cannot output the guessed bit b' in these cases. Thus:

$$\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean}}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean},C1}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean},C2}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean},C3}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda).$$

Case 1. In this case, we show that the advantage that $\mathcal{A}$ has in causing a session π_i^s to set $\pi_i^s.\alpha \leftarrow \texttt{accept}$ when calling $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon)$ and m is not the output of the home network HN is negligible.

Game 0

This game is **Case 1** with cleanness predicate clean in the SUPA experiment as described in Definition 4. Thus we have:

$$\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean},C1}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) = \Pr(break_0).$$

Game 1

In this game we guess the session π_i^s such that π_i^s has reached a status `accept` when calling $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon)$ and m is not an output of the home network HN. Thus we have:

$$\Pr(break_0) = n_S n_N \cdot \big(\Pr(break_1)\big).$$

Game 2

In this game we replace the keys $AKID$, $IKID$ and EK computed in the session π_i^s with uniformly-random values $AKID^*$, $IKID^*$ and EK^* from $\{0,1\}^{|\mathsf{KDF}|}$, where

$|\mathsf{KDF}|$ represents the output length of KDF. Recall that $AKID$, $IKID$, EK are computed honestly as $f_3(K_{ID}, RANDID)$, $f_5(K_{ID}, RANDID)$ and $f_{5^*}(K_{ID}, RANDID)$ respectively. We note that K_{ID} is not used in any other context (and is independent of the long-term key K used in the rest of the 5G AKA), and that for **Case 2** $\mathsf{clean}(\pi_i^s) = \mathtt{true}$, so $\mathcal{A}$ has not issued $\mathbf{Corrupt}(\pi_i^s)$ or $\mathbf{StateReveal}(\pi_i^s)$. Thus, K_{ID} is a uniformly random value that is independent of the protocol execution, and any algorithm that can distinguish Game 2 from Game 1 can be used to construct a simulator $\mathcal{B}$ that distinguishes the output of KDF from random. When the random bit b sampled by the KDF challenger is 0, $AKID = f_3(K_{ID}, RANDID)$, $IKID = f_5(K_{ID}, RANDID)$, $EK = f_{5^*}(K_{ID}, RANDID)$ and $\mathcal{B}$ provides a perfect simulation of Game 1. When $b = 1$, $AKID^*$, $IKID^*$ and $EK^* \xleftarrow{\$} \{0,1\}^{|\mathsf{KDF}|}$ and $\mathcal{B}$ provides a perfect simulation of Game 2. An $\mathcal{A}$ capable of distinguishing Game 2 from Game 1 can therefore break the kdf security of KDF, and thus we have:

$$\Pr(break_1) = \mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \Pr(break_2).$$

Game 3

In this game, we define an abort event $\mathsf{abort}_{\mathsf{dec}}$ that occurs when π_i^s sets $\pi_i^s.\alpha \leftarrow \mathtt{accept}$ during a call to $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon)$ and m is not the output of the home network HN. We do this by constructing a simulator $\mathcal{B}$ that interacts with an AE challenger, computing $\mathsf{AE.Enc}(EK^*, A^+\|B^+)$ by querying $(A^+\|B^+, A^+\|B^+)$ to the LoR AE challenger's $\mathsf{AuthEnc}$ oracle instead of computing it honestly. Similarly, π_i^s decrypts the ciphertext C received in the Update Phase by querying AE challenger with C. Note that EK^* is already uniformly random and independent of the protocol run by Game 3, and this replacement is sound. We note that, by the definition of **Case 1**, $\mathsf{abort}_{\mathsf{dec}}$ must occur. In addition, π_i^s is the first session that sets $\pi_i^s.\alpha \leftarrow \mathtt{accept}$ during a call to $\mathsf{Update}(UE, m, \pi_i^s.st, \epsilon)$ and m is not the output of the home network HN. Moreover, $\mathcal{A}$ cannot terminate and output a guess bit b', and as such the advantage of $\mathcal{A}$ in winning the SUPA experiment in *Game 3* is negligible. Thus we have:

$$\Pr(break_2) = \mathsf{abort}_{\mathsf{dec}}.$$

We now show that the probability of $\mathcal{A}$ causing $\mathsf{abort}_{\mathsf{dec}}$ is negligible. Note that if $\mathsf{abort}_{\mathsf{dec}}$ occurs that $\mathcal{A}$ caused π_i^s to accept when m is not the output of the home network HN. In the proposed protocol, the message m purely consists of $\mathsf{AE.Enc}(EK^*, A^+\|B^+)$. Thus, if $\mathsf{abort}_{\mathsf{dec}}$ occurs, then m is a ciphertext that decrypts correctly by the AE challenger, but was not the output of the query $(A^+\|B^+,$

$A^+ \| B^+$) to the LoR AE challenger's AuthEnc oracle. Thus, when $\mathsf{abort}_{\mathsf{dec}}$ occurs, $\mathcal{B}$ has broken the ae security of the AE challenger, and thus:

$$\mathsf{abort}_{\mathsf{dec}} = \mathsf{Adv}^{\mathsf{ae}}_{\mathsf{AE},\mathcal{A}}(\lambda).$$

Thus we have:

$$\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean},C1}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) = n_N n_S \cdot \big(\mathsf{Adv}^{\mathsf{KDF}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{ae}}_{\mathsf{AE},\mathcal{A}}(\lambda)\big).$$

Case 2. In this case, we show that the advantage that $\mathcal{A}$ has in causing HN to call $\mathsf{Identify}(HN, m, HN.st, K_{HN}) \to (id', m', HN.st')$ such that $\exists\ \pi_i^s.id = id'$, but m was not the output of some $\mathsf{Identify}(UE, \epsilon, \pi_i^s.st, \epsilon)$ and $\mathsf{clean}(\pi_i^s) = \mathtt{true}$ is negligible.

Game 0

This game is **Case 2** with cleanness predicate clean in the SUPA experiment as described in Definition 4. Thus we have:

$$\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean},C2}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) = \Pr(break_0).$$

Game 1

In this game we guess the index of the session π_i^s such that $\pi_i^s.id = id'$ when calling $\mathsf{Identify}(HN, m, HN.st, K_{HN}) \to (id', m', HN.st')$ and m is not the output of π_i^s. Thus we have:

$$\Pr(break_0) = n_N \cdot \big(\Pr(break_1)\big).$$

Game 2

In this game we replace the keys $AKID$, $IKID$ and EK computed in the HN and any stage s of session π_i^s with uniformly-random values $AKID^*$, $IKID^*$ and EK^* from $\{0,1\}^{|\mathsf{KDF}|}$, where $|\mathsf{KDF}|$ represents the output length of KDF. We do so by interacting with n_S KDF challengers. Recall $AKID$, $IKID$, and EK are computed honestly as $f_3(K_{ID}, RANDID)$, $f_5(K_{ID}, RANDID)$ and $f_{5^*}(K_{ID}, RANDID)$ respectively. We note that K_{ID} is not used in any other context (and again, is independent of the long-term key K used in the 5G-AKA), but $RANDID$ is sampled independently in each of the n_S stages, and that for **Case 2** $\mathsf{clean}(\pi_i^s) = \mathtt{true}$, so $\mathcal{A}$ has not issued **Corrupt**(π_i^s) or **StateReveal**(π_i^s).

Thus, K_{ID} is a uniformly random value that is independent of the protocol execution, and such any algorithm that can distinguish Game 2 from Game 1 can be used to construct a simulator $\mathcal{B}$ that distinguishes the output of KDF from random. When the random bit b sampled by the KDF challenger is 0, $AKID = f_3(K_{ID}, RANDID)$, $IKID = f_5(K_{ID}, RANDID)$, $EK = f_{5^*}(K_{ID}, RANDID)$, and $\mathcal{B}$ provides a perfect simulation of Game 1. When $b = 1$, $AKID^*$, $IKID^*$ and $EK^* \xleftarrow{\$} \{0,1\}^{|\mathsf{KDF}|}$ and $\mathcal{B}$ provides a perfect simulation of Game 2. Any $\mathcal{A}$ capable of distinguishing Game 2 from Game 1 can therefore break the kdf security of KDF, and thus we have:

$$\Pr(break_1) = n_S \cdot \mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \Pr(break_2).$$

Game 3

In this game, we define an abort event $\mathsf{abort}_{\mathsf{mac}}$ that occurs when HN outputs $\pi_i^s.id = id'$ during a call to $\mathsf{Identify}(HN, m, HN.st, K_{HN})$ and m is not the output of some stage s of the sessions owned by UE_i. We do this by constructing a simulator $\mathcal{B}$ that interacts with an MAC challenger, computing $\mathsf{MAC}(IKID^*, D\|A\|B\|C)$ by querying $(D\|A\|B\|C)$ to the MAC challenger instead of computing it honestly within HN or any session owned by UE_i. Note that the various $IKID^*$ keys are already uniformly random and independent of the protocol run by Game 2, and this replacement is sound. We note that by the definition of **Case 2** and this case that $\mathsf{abort}_{\mathsf{mac}}$ must occur. In addition, $\mathcal{A}$ cannot terminate and output a guess bit b', and as such the advantage of $\mathcal{A}$ in winning the SUPA experiment in *Game 3* is negligible, and thus we have:

$$\Pr(break_2) = \mathsf{abort}_{\mathsf{mac}}.$$

We now show that the probability of $\mathcal{A}$ causing $\mathsf{abort}_{\mathsf{mac}}$ is negligible. Note that if $\mathsf{abort}_{\mathsf{mac}}$ occurs that $\mathcal{A}$ caused π_i^s to accept when m is not the output of some session owned by UE_i. In PQID, the message m purely consists of $D\|A\|B\|C\|MACID$. Thus, if $\mathsf{abort}_{\mathsf{mac}}$ occurs, then $\mathcal{A}$ has managed to produce a MAC tag under a key $IKID^*$ that verifies correctly, but was not the output of a query to the MAC challenger. Thus, when $\mathsf{abort}_{\mathsf{mac}}$ occurs, $\mathcal{B}$ can forward this to the MAC challenger and break the eufcma security of the MAC, and thus:

$$\mathsf{abort}_{\mathsf{mac}} = n_S \mathsf{Adv}^{\mathsf{eufcma}}_{\mathsf{MAC},\mathcal{A}}(\lambda).$$

Thus we have:

$$\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean},C2}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) = n_N n_S \cdot \big(\mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{eufcma}}_{\mathsf{MAC},\mathcal{A}}(\lambda)\big).$$

Case 3. In this case we show that the advantage that $\mathcal{A}$ has in guessing the test bit b is negligible.

Game 0

This game is **Case 3** with cleanness predicate clean in the SUPA game as described in Definition 4. Thus we have:

$$\mathsf{Adv}^{\mathsf{SUPA},\mathsf{clean},C3}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) = \Pr(break_0).$$

Game 1

In this game we guess the session π_i^s such that $\mathcal{A}$ issues a $\mathbf{Test}(i, s, i', s')$ query and $\pi_i^s = \pi_b$. Thus we have:

$$\Pr(break_0) = n_S n_N \cdot \big(\Pr(break_1)\big).$$

Game 2

In this game we replace the key K_{HN} used in the test session π_i^s with a uniformly random values ${K_{HN}}^*$ from the same distribution $\{0,1\}^\lambda$. We argued at the beginning of §5.7 that $\mathcal{A}$ has negligible chance in detecting this change, and thus:

$$\Pr(break_1) = {}^{q_r}/_{2^{\lambda-1}} + \Pr(break_2).$$

Game 3

In this game we replace the value $CKID^+$ computed in the previous stage of the test session π_i^{s-1} with a uniformly-random value $CKID^{+*}$ from $\{0,1\}^{|\mathsf{KDF}|}$ where $|\mathsf{KDF}|$ represents the output length of KDF. Note that $CKID^+$ is computed honestly as $f_4({K_{HN}}^*, K_N)$ respectively. We note that K_{HN} is not used in any other context, and that for **Case 3** $\mathsf{clean}(\pi_i^s) = \mathtt{true}$, so $\mathcal{A}$ has not issued $\mathbf{Corrupt}(\pi_i^s)$ or $\mathbf{StateReveal}(\pi_i^s)$. Thusly, ${K_{HN}}^*$ is a uniformly random value that is independent of the protocol execution, and as such any algorithm that can distinguish Game 3 from Game 2 can be used to construct a simulator $\mathcal{B}$ that distinguishes the output of KDF from random. When the random bit b sampled by the KDF challenger is 0, $CKID^+ = f_4({K_{HN}}^*, K_N)$, and $\mathcal{B}$ provides a perfect simulation of Game 2. When

$b = 1$, $CKID^{+*} \xleftarrow{\$} \{0,1\}^{|\mathsf{KDF}|}$ and $\mathcal{B}$ provides a perfect simulation of Game 3. An $\mathcal{A}$ capable of distinguishing Game 3 from Game 2 can therefore break the kdf security of KDF, and thus we have:

$$\Pr(break_2) = \mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \Pr(break_3).$$

Game 4

In this game we replace the keys $AKID$, $IKID$ and EK computed in the *previous stage* of the test session π_i^{s-1} with uniformly-random values $AKID^*$, $IKID^*$ and EK^* from $\{0,1\}^{|\mathsf{KDF}|}$ where $|\mathsf{KDF}|$ represents the output length of KDF. Note that $AKID, IKID, EK$ are computed honestly as $f_3(K_{ID}, RANDID^*)$, $f_5(K_{ID}, RANDID^*)$ and $f_{5^*}(K_{ID}, RANDID^*)$ respectively. We note that K_{ID} is not used in any other context (again, K_{ID} is independent of the long-term key K used in the 5G-AKA), and that for **Case 3** $\mathsf{clean}(\pi_i^s) = \mathtt{true}$, so $\mathcal{A}$ has not issued $\mathbf{Corrupt}(\pi_i^s)$ or $\mathbf{StateReveal}(\pi_i^s)$.

Thus, K_{ID} is a uniformly random value that is independent of the protocol execution, and any algorithm that can distinguish Game 4 from Game 3 can be used to construct a simulator $\mathcal{B}$ that distinguishes the output of KDF from random. When the random bit b sampled by the KDF challenger is 0, $AKID = f_3(K_{ID}, RANDID^*)$, $IKID = f_5(K_{ID}, RANDID^*)$, $EK = f_{5^*}(K_{ID}, RANDID^*)$, and $\mathcal{B}$ provides a perfect simulation of Game 3. When $b = 1$, $AKID^*$, $IKID^*$ and $EK^* \xleftarrow{\$} \{0,1\}^{|\mathsf{KDF}|}$ and $\mathcal{B}$ provides a perfect simulation of Game 4. An $\mathcal{A}$ capable of distinguishing Game 4 from Game 3 can therefore break the kdf security of KDF, and thus we have:

$$\Pr(break_3) = \mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \Pr(break_4).$$

Game 5

In this game, we replace the A^+, B^+ values sent from HN to the test session's previous stage π_i^{s-1} with uniformly-random values A^{+*}, B^{+*}. We do this by constructing a simulator $\mathcal{B}$ that interacts with an AE challenger, computing $\mathsf{AE.Enc}(EK^*, A^+\|B^+)$ by querying $(A^+\|B^+, A^{+*}\|B^{+*})$ to the LoR AE challenger's $\mathsf{AuthEnc}$ oracle instead of computing it honestly. Similarly, π_i^{s-1} decrypts the ciphertext C received in the *update phase* by simply querying it to the $\mathsf{AuthDec}$ oracle, ignoring the result of the decryption and using the A^+, B^+ values created by HN instead. Note that EK^* is already uniformly random and independent of the protocol run by Game 4, and

this replacement is sound. We note that this means that when the bit b sampled by the AE challenger is 0, then A^+, B^+ values are sent honestly, but when the bit b sampled by the AE challenger is 1, then A^{+*}, B^{+*} values are sent instead and A^+ and B^+ are established independently of the ciphertext sent during the Updation Phase. Any adversary capable of distinguishing this change can be turned into an adversary capable of breaking the security of the AE scheme, and thus we have:

$$\Pr(break_4) = \mathsf{Adv}^{\mathsf{ae}}_{\mathsf{AE},\mathcal{A}}(\lambda) + \Pr(break_5).$$

Game 6

In this game we replace the value $CKID$ computed in the test session π_i^s with a uniformly-random value $CKID^*$ from $\{0,1\}^{|\mathsf{KDF}|}$ where $|\mathsf{KDF}|$ represents the output length of KDF. Note that $CKID$ is computed honestly as $f_4({K_{HN}}^*, K_N)$. We note that K_{HN} is not used in any other context, and that for **Case 3** $\mathsf{clean}(\pi_i^s) = \mathtt{true}$, so $\mathcal{A}$ has not issued $\mathbf{Corrupt}(\pi_i^s)$ or $\mathbf{StateReveal}(\pi_i^s)$. Thus, ${K_{HN}}^*$ is a uniformly random value that is independent of the protocol execution, and as such any algorithm that can distinguish Game 3 from Game 2 can be used to construct a simulator $\mathcal{B}$ that distinguishes the output of KDF from random. When the random bit b sampled by the KDF challenger is 0, $CKID = f_4({K_{HN}}^*, K_N)$, and $\mathcal{B}$ provides a perfect simulation of Game 5. When $b = 1$, $CKID^* \xleftarrow{\$} \{0,1\}^{|\mathsf{KDF}|}$ and $\mathcal{B}$ provides a perfect simulation of Game 6. An $\mathcal{A}$ capable of distinguishing Game 6 from Game 5 can therefore break the kdf security of KDF, and thus we have:

$$\Pr(break_5) = \mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \Pr(break_6).$$

Game 7

In this game we replace the keys $AKID$, $IKID$ and EK computed in the session π_i^s with uniformly-random values $AKID^*$, $IKID^*$ and EK^* from $\{0,1\}^{|\mathsf{KDF}|}$ where $|\mathsf{KDF}|$ represents the output length of KDF. Note that $AKID$, $IKID$, EK are computed honestly as $f_3(K_{ID}, RANDID)$, $f_5(K_{ID}, RANDID)$ and $f_{5^*}(K_{ID}, RANDID)$ respectively. We note that K_{ID} is not used in any other context and also that $RANDID$ is independent from the $RANDID^*$ sampled in the previous stage, and that for **Case 3** $\mathsf{clean}(\pi_i^s) = \mathtt{true}$, so $\mathcal{A}$ has not issued $\mathbf{Corrupt}(\pi_i^s)$ or $\mathbf{StateReveal}(\pi_i^s)$.

Thus, K_{ID} is a uniformly random value that is independent of the protocol execution, and any algorithm that can distinguish Game 7 from Game 6 can be used to construct

a simulator $\mathcal{B}$ that distinguishes the output of KDF from random. When the random bit b sampled by the KDF challenger is 0, $AKID = f_3(K_{ID}, RANDID)$, $IKID = f_5(K_{ID}, RANDID)$, $EK = f_{5^*}(K_{ID}, RANDID)$ and $\mathcal{B}$ provides a perfect simulation of Game 6. When $b = 1$, $AKID^*$, $IKID^*$ and $EK^* \xleftarrow{\$} \{0,1\}^{|\mathsf{KDF}|}$ and $\mathcal{B}$ provides a perfect simulation of Game 7. An $\mathcal{A}$ capable of distinguishing Game 6 from Game 7 can therefore break the kdf security of KDF, and thus we have:

$$\Pr(break_6) = \mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \Pr(break_7).$$

Game 8

In this game we replace $SQNID_{UE}$ in both the test session and its matching home network HN with a uniformly-random value from the same distribution. Note that $SQNID_{UE}$ sent once during the protocol execution: as the first field $D = SQNID_{HN} \oplus AKID^*$ in the $SUCI$ message during the Identification Phase. By *Game 7*, $AKID^*$ is a uniformly-random value independent of the protocol execution, and thus we can consider it a one-time-pad to $SQNID_{UE}$, perfectly hiding it. As a result, we can replace $SQNID_{UE}$ with any value without detection. In addition, in any future sessions π_i^{s+1} we instead increment $SQNID_{UE}$ from session π_i^{s-1} instead of incrementing from π_i^s. Thus we have:

$$\Pr(break_7) = \Pr(break_8).$$

Game 9

In this game, we replace the A^+, B^+ values sent from HN to the test session's previous stage π_i^s with uniformly-random values A^{+*}, B^{+*}. We do this by constructing a simulator $\mathcal{B}$ that interacts with an AE challenger, computing $\mathsf{AE.Enc}(EK^*, A^+\|B^+)$ by querying $(A^+\|B^+, A^{+*}\|B^{+*})$ to the LoR AE challenger's $\mathsf{AuthEnc}$ oracle instead of computing it honestly. Similarly, π_i^s decrypts the ciphertext C received in the *update phase* by simply querying it to the $\mathsf{AuthDec}$ oracle, ignoring the result of the decryption and using the A^+, B^+ values created by HN instead. Note that EK^* is already uniformly random and independent of the protocol run by Game 7, and this replacement is sound. We note that this means that when the bit b sampled by the AE challenger is 0, then A^+, B^+ values are sent honestly, but when the bit b sampled by the AE challenger is 1, then A^{+*}, B^{+*} values are sent instead and A^+ and B^+ are established independently of the ciphertext sent during the *update*

phase. Any adversary capable of distinguishing this change can be turned into an adversary capable of breaking the security of the AE scheme, and thus we have:

$$\Pr(break_8) = \mathsf{Adv}^{\mathsf{ae}}_{\mathsf{AE},\mathcal{A}}(\lambda) + \Pr(break_9).$$

We argue now that all values sent in the tested session π_i^s are independent from any value sent in previous and future sessions. Thus we have:

$$\Pr(break_9) = 0.$$

Thus we can show:

$$\mathsf{Adv}^{\mathsf{SUPA,clean},C3}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) \leq n_N n_S \cdot \left(q_r/2^{\lambda-1} + 4 \cdot \mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{ae}}_{\mathsf{AE},\mathcal{A}}(\lambda)\right).$$

Summing the previous cases allows us to show:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{SUPA,clean}}_{\mathsf{PQID},n_N,n_S,\mathcal{A}}(\lambda) \leq\ & n_N n_S \cdot \left(\mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{ae}}_{\mathsf{AE},\mathcal{A}}(\lambda)\right) \\ & + n_N n_S \cdot \left(\mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{eufcma}}_{\mathsf{MAC},\mathcal{A}}(\lambda)\right) \\ & + n_N n_S \cdot \left(q_r/2^{\lambda-1} + 4 \cdot \mathsf{Adv}^{\mathsf{kdf}}_{\mathsf{KDF},\mathcal{A}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{ae}}_{\mathsf{AE},\mathcal{A}}(\lambda)\right).\end{aligned}$$

5.7.2 Other Improvements

We now discuss how our proposal prevents certain attacks and motivate our proposals to change aspects of the 3GPP specification.

- **Update of Long-Term Secret Parameters.** As elaborated in §5.4.2, it may be required for HN to update its long-term secret key. In the current ECIES-based mechanism this is a difficult proposition as it requires a suitable mechanism to transport the updated public key of the HN to all of its subscribers and also an update-confirmation mechanism used by the subscribers. With our proposal, no such mechanism is required as the secret key is internal to HN. However, updating the K_{HN} will require an interim period during which the HN has to operate with both the new and old key, but this would be handled within domains of the *identification phase* (§5.5.2) itself.

- **Migration to Authenticated Encryption in 5G.** Our proposal uses authenticated encryption to update identification parameters. Currently, the 3GPP specifications do not list authenticated encryption algorithms, but instead separate encryption and integrity algorithms, ascribed to historical reasons. Previous generations of mobile telephony used to avoid integrity protection of user traffic (voice/data) because of the substantial errors during the

radio channel propagation. Only the signalling traffic used to be integrity protected. But as the quality of radio traffic improved, provisions for integrity protection of user traffic were also created. Though we could have achieved the requisite security guarantees in our scheme using the currently specified primitives by following the "Encrypt-then-authenticate" paradigm, we stress that our approach is clearer and suggest that the 3GPP specifications should introduce such primitives.

- **Replay Prevention.** We include and authenticate sequence numbers *SQNID* in our protection scheme to prevent replay attacks. Moreover, they also provide appropriate resilience to desynchronization between the UE and HN as now an arbitrary third party cannot initiate an identification request without access to the shared secret key K_{ID}.

- **Chosen SUPI Attacks.** Our scheme is resilient to *chosen SUPI attacks* (§5.4.2), due to inclusion of the shared secret key K_{ID} as the keying input for the computation of the MAC tag *MACID*.

- **Multiple Identification Parameters.** In the case of an unexpected interruption, the UE will re-attempt identification using the same parameters A and B. Although this does not violate the session unlinkability criterion (as it is effectively the same session), one could imagine the UE storing multiple pairs of identification parameters in these cases.

5.8 Parameter Sizes

For 5G, the most cryptographically relevant quantum algorithms are Grover's searching algorithm [68] (quadratically faster than any classical brute force searching scheme) and Shor's factoring algorithm [130] (exponentially faster than the best known classical factoring algorithm - the number field sieve). It is worth noting, however, that if an alternative was suggested that utilizes the symmetric-key primitives offered by the current 3GPP specification (and their associated parameter sizes), then this may not achieve quantum security. For example, the output of the MAC algorithm (referred to as f_1, see Table 5.2) could be 64 bits. For such a proposal to realize resilience against quantum algorithms [45], the standard technique to achieve is to increase the length of the classical-secure key size, preferably to 256 bits. In this regard, 3GPP is already working towards supporting 256-bit algorithms [21]. As regards the effects of bidding down attacks, in the current 3GPP

specifications the *SUPI* is derived directly from the IMSI, and is thus susceptible to bidding down attacks (§5.4.2) by an active adversary. To thwart such attacks, the derivation of *SUPI* should be independent of the previous generations' IMSI.

5.9 Chapter Summary

In this chapter we introduced a new private identification scheme for the 5G specification, a quantum-secure alternative to the current public-key based solution. We described the limitations of the existing solution and discussed how our proposal mitigates these drawbacks. We introduced a security model for our protocols and proved the security of our proposal. Our proposal is compatible with the current 5G specifications, depending mostly on cryptographic primitives already specified in 5G, adding minimal performance overhead and requiring minor changes in the existing message structure.

Chapter 6

Mitigating Downgrade Attacks on 5G

The 5G Release 15 SUPI protection mechanism (§5.4.1) and our alternative SUPI protection scheme PQID *(§5.5) are both vulnerable to downgrade attacks, i.e. an active attacker is able to force the connection down to 2G/3G/4G and exploit previously known vulnerabilities. In this chapter we show how a recent downgrade protection proposal for 5G Release 15 can be amalgamated with our SUPI protection scheme to provide a quantum-secure and downgrade-resistant private identification mechanism for 5G.*

6.1 Introduction and Background

Despite the SUPI protection provisioned by 3GPP Release 15 (§5.4.1), IMSI-catching remains unaffected in GSM/UMTS/LTE networks, which opens the possibility of a downgrade attack on 5G user privacy. When a 5G UE wants to connect to a GSM/UMTS/LTE SN, the SN wants to know the identity of the user so that the user can be billed. The SN can send an IMSI inquiry to the UE. Since the link between the UE and the SN is initially unprotected, the UE has to respond with its IMSI in plaintext. A passive IMSI-catcher can listen to the radio channel and read IMSIs sent in plaintext. An active IMSI-catcher can impersonate a legitimate SN and can make an IMSI inquiry. The UE has no way to distinguish an active IMSI-catcher from a legitimate SN before authenticating the SN. Hence, the UE invariably has to send the plaintext IMSI to the attacker, otherwise it will be locked out of the network. As in 5G networks, this identification is followed by mutual authentication between UE and SN based upon a challenge-response based AKA protocol.

Khan et al. [89] looked at this issue in 5G networks and proposed a solution to

countermeasure downgrade attacks. Although we will only consider LTE networks in our discussion about downgrade attacks, our analysis is equally applicable to UMTS networks. As GSM networks lack mutual authentication between UE and HN, this solution is not applicable to GSM networks in a straightforward manner. The solution in [89] basically proposes the following:

- **When interacting with LTE networks:** To use an existing pseudonym-based solution [90] to protect identity of 5G users against IMSI-catchers.
- **When interacting with 5G networks:** To include a mechanism for updating the LTE pseudonyms using the 5G ECIES-based identity protection scheme.

The update mechanism helps in recovering from an (unlikely) loss of pseudonym synchronization between a UE and its HN once they reconnect over a 5G network. The proposed solution utilizes existing LTE messages and requires minimalistic changes to the 3GPP Release 15 messages. It requires modifications only in the UE and HN (not within SN) in order to provide identity privacy. The solution uses pseudonyms that have the same format as LTE IMSIs to defeat the downgrade attack. A pseudonym looks like a normal LTE IMSI, but its MSIN part is randomized and frequently changing. The UE is provisioned with two pseudonyms at the beginning and it obtains fresh pseudonyms during further AKA protocol runs. It uses these pseudonyms instead of IMSI to identify itself when connecting to an LTE SN. This solution piggybacks on existing messages involved in the LTE-AKA and 5G-AKA protocols to deliver new pseudonyms to the UE and does not require separate messages.

Since the total number of pseudonyms is limited, pseudonyms need to be reused, i.e., disassociated from one user and reallocated to another. In this solution the UE updates its pseudonyms by interacting with HN via either an LTE SN or 5G SN. The UE does not need a pseudonym to connect to a 5G SN. Hence, even in the unlikely event where the UE and HN lose synchronization of pseudonyms, the synchronization can be restored by obtaining a new LTE pseudonym from the HN, simply by connecting to a 5G SN.

The work in this chapter shows how the downgrade protection proposal by Khan et al. [89], which was originally for 5G Release 15, can be combined with our SUPI protection scheme PQID (Chapter 5) to provision both quantum-security and

downgrade-resistance for private identification in 5G. The rest of the chapter is organized as follows: §6.2 details the proposal by Khan et al, 6.3 presents an analysis of the downgrade protection proposal, while §6.4 shows the amalgamation of the downgrade proposal with our PQID. §6.5 discusses the combined solution and §6.6 provides the summary.

6.2 The Downgrade Protection Solution

We now explain the details of the 5G downgrade protection solution as proposed by [89]. We use similar notation to that used in [89] for ease of comparison. In this solution, instead of IMSI, a 5G UE when interacting with an LTE network uses pseudonyms that have the same format as an LTE IMSI. In addition, when a 5G UE runs the 5G-AKA, it synchronizes its LTE pseudonyms with the HN. In this solution:

- A 5G UE uses pseudonyms to connect with LTE SNs and SUCI to connect with 5G SNs.
- Only the HN allocates and releases pseudonyms of mobile users; initially the HN allocates two pseudonyms per 5G user and provisions them into the users USIMs.
- A 5G UE obtains new pseudonyms by participating in authentication protocols (LTE-AKA or 5G-AKA). The two latest pseudonyms received by the UE are denoted by p_1 and p_2.
- In order to support simultaneous connections with multiple 5G SNs, the solution:
  - Uses a subscriber-specific counter d of pseudonyms maintained by the HN.
  - Keeps track of in-use pseudonyms in 5G UE and HN, using sets P_{UE} and P_{HN}, respectively; the elements of these sets are pairs (p_i, d_i) of pseudonyms and their respective counters (see Figure 6.1).
  - Transmits information about pseudonyms being used in 5G UE via the SUCI.

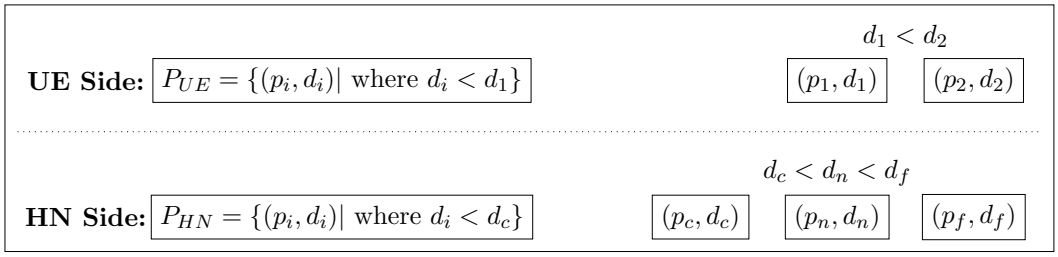


Figure 6.1: Pseudonym state in UE and HN.

The 5G UE uses only p_1 or p_2 when replying to an IMSI inquiry from an LTE SN. The set P_{UE} contains pseudonyms that 5G UE received before p_1 and p_2. The UE deletes pseudonyms from P_{UE} based on policy provided by the HN that could include, for example, pseudonyms' lifetime or the maximum size of P_{UE}. The P_{HN} contains pseudonyms that the HN thinks are in P_{UE}, and it deletes pseudonyms from P_{HN} according to another policy. One objective of these policies is that the HN should not delete a pseudonym which the UE has not yet deleted. Thus, P_{UE} is a subset of P_{HN}. In short, the UE informs the HN about its older pseudonyms when it connects with a 5G SN using SUCI, and then the HN is able to reduce the set P_{HN}. Note that as long as the UE is connecting to LTE SN only, the size of P_{HN} grows.

This solution does not introduce any new messages on top of what 3GPP has standardized. Although the authors of [89] claim that their scheme is completely transparent to a participating 5G SN, it is not. It modifies the length of a couple of existing 5G-AKA messages (for details see Steps (3) and (4) of Figure 6.3) while the LTE-AKA messages remain fully transparent to the SN. It only introduces changes in the 5G UE and its HN. However, to enable *Lawful Interception*, this solution does require some modifications within the LTE SNs.

As standardized, a 5G USIM comes with an IMSI, a master key K and the HN's public key pk embedded in it. In this solution, a 5G USIM is also provisioned with two pseudonyms p_1, p_2 and a key k, shared with HN for decrypting the pseudonyms. Similarly, along with the user's IMSI, and master key K, the HN in this solution has to store additional information: the shared key k for encrypting pseudonyms and three pseudonyms p_c, p_n, and p_f (where the subscripts stand for current, next and future). Ideally $p_1 = p_c$ and $p_2 = p_n$. When a pseudonym p is allocated to a subscriber, it is associated with a subscriber-specific counter d, which is a strictly monotonically increasing counter value that increments each time the HN allocates a new pseudonym to the subscriber.

6.2.1 LTE-AKA based Solution

This part of the solution, when the 5G UE interacts with an LTE SN, is shown in Figure 6.2 and works as follows:

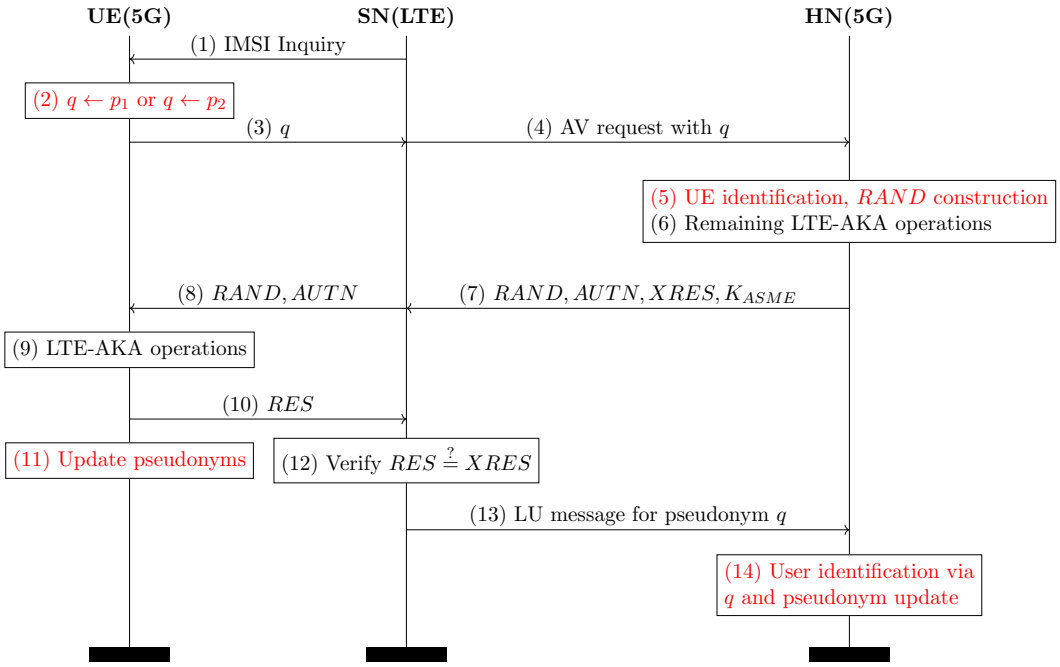


Figure 6.2: LTE-AKA based solution. The differences to the standard LTE-AKA are highlighted in red.

1. An LTE SN requests the IMSI from the UE.
2. The UE chooses one of the pseudonyms p_1, p_2 and assigns it to q.
3. The UE sends q to the SN.
4. The SN sends an AV request for the pseudonym q to the HN. The user mostly identifies itself with the GUTI. Sometimes the user may implicitly identify itself by responding to a paging message. In either case, if the SN wants to perform an LTE-AKA, the SN requests an AV from the HN for the pseudonym/IMSI that was associated with the GUTI or was used in the paging message.
5. The HN checks whether the pseudonym q is in use for any subscriber. If it is, the HN starts to prepare the AV. It first constructs the random challenge RAND. A new pseudonym is embedded in the RAND as follows:
   - The 128-bit long random challenge RAND is created by encrypting (using key k that can be generated from the master key K) the pseudonym p_f,

its counter d_f, an error correction flag (ECF) and a randomly chosen l-bit long salt. If the pseudonym p_f is null, a new m-bit long p_f is chosen randomly from the pool of unused pseudonyms. Then d_f is set to the current value of the counter CTR, which is a strictly monotonically increasing counter maintained by the HN. It increases each time the HN generates a new pseudonym. The flag ECF is by default set to 0 but a 5G HN may set it to some other values to notify the UE about an error in the UEs pseudonym state.

- The length of l is equal to $(128 - length(d_f) - length(ECF) - m)$ bits. The length of m depends on how many digits of the IMSI are randomized. Since the number of randomized digits can be at most 10, $m \leq 34$. The length of d_f and ECF depends on implementation; length$(d_f) \leq 24$ and length$(ECF) \leq 2$ bits should be enough. This implies length$(l) \geq 68$.

6. The HN computes other parts of the authentication vector AV (in addition to the RAND), including the expected response XRES to the challenge RAND, anchoring key K_{ASME}, and authentication token AUTN [16].

7. The HN sends RAND, AUTN, XRES and K_{ASME} to the SN.

8. The SN forwards RAND and AUTN to the UE.

9. The UE performs LTE-AKA related operations, like verifying the MAC in AUTN and computing the response RES [16].

10. If everything is fine in Step (9), the UE sends RES to the SN.

11. The UE decrypts RAND to extract the embedded pseudonym p, and the counter d, and updates the pseudonyms p_1, p_2 if p is new. These operations are as follows:

    - UE decrypts RAND using key k and obtains p, d, ECF and salt.
    - In an LTE-AKA, the ECF field is always set to 0.
    - If the pseudonym p is a new pseudonym, i.e., $d > d_2$, then the UE inserts (p_1, d_1) into P_{UE} and sets (p_1, d_1),$(p_2, d_2) \leftarrow (p_2, d_2)$,$(p, d)$.
    - If $d \leq d2$, then p is considered as an old pseudonym and the UE does not update pseudonyms. If somehow the value of d_2 (in the UE) gets corrupted and becomes larger than d_f (in the HN), the UE would no longer be able to accept new pseudonyms just by running LTE-AKA. In this case, the UE can still obtain a new pseudonym by running a 5G-AKA.

12. The SN compares RES and XRES and aborts if RES $\neq$ XRES.
13. The SN sends a Location Update (LU) message to the HN for the pseudonym q.
14. The HN searches for a user $q \in \{p_n, p_f\}$. If found, and p_f is not null, then the HN inserts (p_c, d_c) into P_{HN} and sets (p_c, d_c),(p_n, d_n),(p_f, d_f) $\leftarrow$ (p_n, d_n),(p_f, d_f),(NULL, NULL).

6.2.2 5G-AKA based Solution

The following are the major aims of this phase of the solution:

- Deliver a new pseudonym to a 5G UE using the 5G-AKA;
- Notify the HN about pseudonyms that the UE is not using any more so that those pseudonyms can be reused by the HN;
- Re-synchronize pseudonym states in the (rather unlikely) erroneous situation where d_2 becomes greater than d_f.

It is required that new pseudonyms be delivered to a 5G UE even when the UE has not used the existing pseudonyms to connect with a legitimate LTE SN. This is because the UE may have used those pseudonyms with an active IMSI-catcher. If a 5G UE always connects with a 5G SN and does not get new pseudonyms by participating in the 5G-AKA, then the 5G UE will have the same pseudonyms for a long time. If an active IMSI-catcher makes many IMSI inquiries over this time then the UE would respond to each of those IMSI inquiries with the same pseudonym. Thus, an active IMSI-catcher would be able to track and monitor the user with this long-lived pseudonym.

This solution is built on the 5G-AKA protocol of Release 15 [14] with changes only in the 5G UE and the HN. The solution is thus mostly transparent to the 5G SNs of Release 15 except for the length of a couple of messages (Steps (3) and (4)). This part of the solution requires the HN and the USIM to contain all the information of the LTE-AKA based part of the solution. Moreover, it requires the HN to have a public/private key pair pk, sk and the USIM to be provisioned with the HNs public key pk. This solution is presented in Figure 6.3 and proceeds as follows:

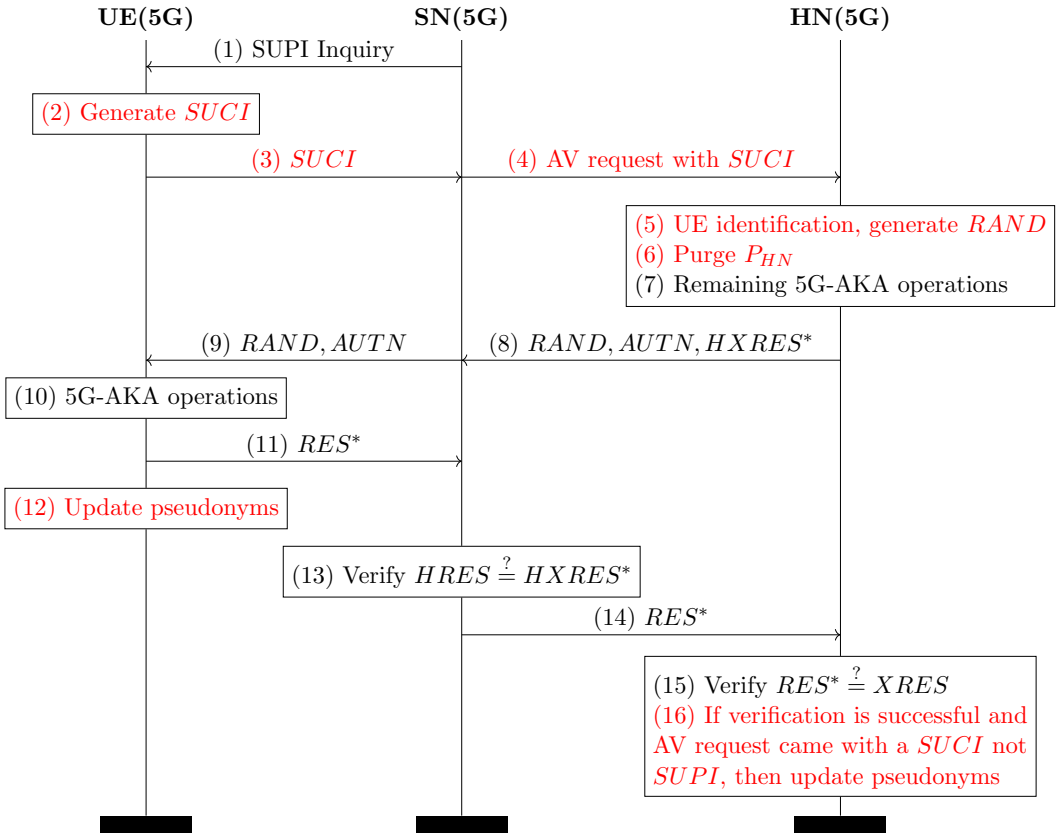


Figure 6.3: 5G-AKA based solution. The differences to 5G-AKA are highlighted in red.

1. A 5G SN requests the SUPI from the UE.

2. The UE generates a SUCI. Here, the plaintext encrypted into SUCI is different from that of Release 15 (see §5.4.1). Two counters, δ_{min} and δ_{max}, are also encrypted along with the MSIN. δ_{min} is the smallest counter of all the pseudonyms in the UE. The HN can thus know which pseudonyms the UE is no longer using. Consequently, they can be allocated to other UEs. The value of the counter δ_{max} is always set to d_2. The construction of the SUCI is as follows:

   - UE computes MAC T of message $MSIN||\delta_{min}||\delta_{max}$ with master key K.
   - UE encrypts $MSIN||\delta_{min}||\delta_{max}||T$ with HNs public key pk.
   - The ciphertext is concatenated with other plaintext like HN identifier, the public-key identifier of the HN, and the SUPI protection scheme identifier. The outcome is returned as the SUCI.

3. The UE sends the SUCI to the SN.

4. The SN forwards the SUCI to the HN, requesting an AV. Most of the time the user identifies with the GUTI. Sometimes the user may implicitly identify itself by responding to a paging message. In either case, if the SN wants to perform a 5G-AKA, the SN requests an AV from the HN with the SUPI that was associated with the GUTI or paging.

5. The HN constructs the RAND by embedding a pseudonym in it as follows:

   - The HN extracts $MSIN, \delta_{min}, \delta_{max}$ and T from the encrypted part of the SUCI using the private key sk.
   - HN verifies the MAC T using master key K. If this verification fails, the HN aborts.
   - HN checks if p_f is NULL. If yes, an m-bit long p_f is randomly allocated (from the pool of free pseudonyms) and d_f is set to CTR, which is a subscriber-specific counter maintained by the HN. It increases every time the HN generates a new pseudonym.
   - HN checks whether δ_{max} is greater than d_f . If yes, it sets ECF to 1, otherwise ECF is set to 0.
   - A l-bit long random $salt$ is chosen.
   - $(p, d) \leftarrow (p_f, d_f)$.
   - $(p, d, ECF, salt)$ is encrypted with key k. The resultant ciphertext is $RAND$.

6. Further, HN removes pseudonyms from P_{HN} which have counter smaller than δ_{min}.

7. HN performs the other operations of 5G-AKA, except the construction of $RAND$.

8. HN then sends an AV $(RAND, AUTN, HXRES^*)$ to the SN.

9. SN forwards the RAND and AUTN to the UE.

10. UE performs the 5G-AKA related operations.

11. UE then sends the response RES* to the SN.

12. UE further decrypts RAND, extracts the embedded pseudonym from the RAND, and updates the pseudonyms in the UE. In 5G-AKA, the ECF might be set to 1 by the HN. In this case the UE will empty the set P_{UE}, set

$(p_1, d_1), (p_2, d_2) \leftarrow (p, d-1), (p, d)$ and terminate the algorithm at this step. This is needed to recover from a very unlikely error situation where d_2 gets corrupted in the UE.

13. SN computes HRES* as a function of RES* and then compares HRES* with HXRES*.

14. If the comparison in Step (13) matches, SN forwards the RES* to the HN.

15. The HN compares RES* and XRES.

16. If the comparison in Step (15) matches, HN checks whether the AV (associated with the current 5G-AKA run) came with a SUCI or a SUPI. If with a SUCI, HN checks if the pseudonym p that was embedded in the RAND is still p_f. If yes, the HN moves (p_c, d_c) to P_{HN} and sets $(p_c, d_c), (p_n, d_n), (p_f, d_f) \leftarrow (p_n, d_n), (p_f, d_f), (NULL, NULL)$. Consequently, the HN would embed a new pseudonym in the RAND while responding to the next AV request. If the UE identified itself with GUTI or responded to a paging message, then the subsequent AV request sent by the SN would be with a SUPI. Hence, the pseudonyms would not get updated in HN. This means that in response to the next AV request, the HN will embed the same pseudonym in the RAND.

Step (16) helps the system to avoid generating unnecessary pseudonyms. If a 5G UE attempts to connect with an LTE SN using a pseudonym, but the subsequent LTE-AKA fails or no LTE-AKA follows (possibly because the SN is an active IMSI-catcher), then the next time the UE tries to connect with a 5G SN it can use SUCI instead of GUTI. In this way the UE can notify the HN that it needs a new pseudonym, and it will receive a new pseudonym in the next AKA. Thus, the solution avoids generating unnecessary pseudonyms.

6.2.3 Pseudonym Allocation and Removal Process

The HN randomly allocates a new pseudonym for a user from a pool of free pseudonyms. A pseudonym p can be in the pool of free pseudonyms only if it is not in the set P_{HN} (or used as p_c, p_n or p_f) for any user. As new pseudonyms are generated for a user, the older pseudonyms are stored in the sets P_{HN} and P_{UE}. A pseudonym should not be allocated to a new user as long as it is in P_{HN} and P_{UE} of any other user. If pseudonyms are never removed from P_{HN} and P_{UE}, the system will eventually run out of free pseudonyms. To keep the pool of free pseudonyms

large enough, the HN needs to remove old pseudonyms that are no longer used by a UE from P_{HN}. Hence, a policy is required for removal of pseudonyms from these sets. One objective of this policy should be that a pseudonym should not be deleted from P_{HN} of a user that is not yet deleted from the set P_{UE} of that user.

In this downgrade protection solution, a UE removes pseudonyms from P_{UE} according to the policies provisioned in the UE by the HN. The UE notifies the HN about the pseudonyms that the HN can remove from the P_{HN}. The UE sends (encrypted in the SUCI message) the smallest counter value δ_{min} of all the pseudonyms available in the UE. The HN then removes (from P_{HN}) all pseudonyms that have smaller counter values than δ_{min} . The UE sends δ_{min} with both integrity and confidentiality protection within the encrypted part of SUCI, as discussed in §6.2.2.

It is important to define when the UE can decide that it no longer uses a pseudonym and it can be removed from P_{UE}. The pseudonyms in P_{UE} are stored because the UE should be able to respond to paging messages sent by the SN. Therefore, if a UE has a pseudonym (and the associated GUTI) that has not been used for a reasonably long time (as defined in the policy) and the UE is currently connected to a different SN, the pseudonym can be removed.

The UE may have an old pseudonym in P_{UE} that is associated with a GUTI and a security context but has no other pseudonyms associated with the same SN and the UE is currently connected to this SN. In such a case, the UE would initiate a new registration procedure with the SN using pseudonym p_1 or p_2. If this registration is successful, the UE can remove the old pseudonym from P_{UE}. The UE may also follow a guideline set by the HN to remove pseudonyms from P_{UE}; for example, removing pseudonyms that are older than one day.

6.3 Analysis of the Proposed Solution

In the downgrade protection proposal by [89], new pseudonyms are delivered to the UE under confidentiality protection of the key k. Therefore, an LTE-based IMSI catcher cannot know a pseudonym before the UE uses it. This ensures unlinkability between various pseudonyms being utilized by the same user. We further analyze this downgrade protection proposal now.

6.3.1 Pseudonym Synchronization

Desynchronization between UE and HN means that the pseudonyms p_1 and p_2 associated a user within the UE are no longer associated with same user within the HN. We say that a UE is desynchronized with its HN if the following condition occurs:

$$(p_1, d_1), (p_2, d_2) \notin \{(p_c, d_c), (p_n, d_n), (p_f, d_f)\} \cup P_{HN}.$$

In this downgrade protection mechanism, if both UE and HN function correctly, a desynchronization scenario cannot occur. However, if due to some unlikely error (such as HN memory erasure, etc.) desynchronization occurs, the UE can easily resynchronize by connecting to a 5G SN. This is because, to participate in a 5G-AKA, there is no requirement for pseudonyms and, after a successful 5G-AKA, the UE will be allotted new valid pseudonyms.

6.3.2 Lawful Interception

The 3GPP LI requirements state that a network operator should be able to intercept communications independently without the need to rely on another network operator or party. In particular, a SN should not be required to provide the identity of the target user to the HN and vice versa [10, 25]. In 5G Release 15, LI is ensured via two features:

1. The HN provides the SUPI of the user to the SN during the 5G-AKA (see §2.7). In this way, even if a SUCI was utilized by the user for initial identification, the SN comes to know of the actual long-term identifier of the user before provisioning actual services.
2. Both UE and SN use SUPI as one of the inputs to derive the master session key [14]. This ensures that communication can only occur if both parties agree on the same SUPI.

The authors of [89] propose LI in their solution by suggesting to incorporate Feature 1: i.e., after a successful LTE-AKA, the HN should provision the IMSI of the user to the SN as part of the LTE-AKA protocol. This of course will require software updates in both LTE SN and HN. Incorporating Feature 2 is more involved and will require the MSIN part of the user's IMSI as an additional input during the derivation of the master session key by UE and SN.

6.3.3 Performance Overheads

In this downgrade protection solution, the pseudonym delivery and update is undertaken via existing (LTE/5G) AKA messages. The overall structure of these existing messages remains unchanged, however parts of these messages are constructed and interpreted differently. Hence, there is no communication overhead and delay as existing message lengths remain the same (except for a minor increase in two messages of the 5G-AKA) with no extra messages. There is some additional computational overhead involved, such as during embedding of pseudonyms within the RAND, etc. However, most of these additional operations are carried out in the HN and consist of symmetric cryptographic operations, hence the impact is negligible.

6.4 Quantum Security with Downgrade Resistance

In this section we present the details of the combining of our PQID proposal (§5.5) with the downgrade protection proposal (§6.2) to come up with a 5G user identification mechanism which is both quantum-secure and downgrade resistant. Basically, the aim is to replace the public-key based operations of §6.2 with the symmetric-key based operations of §5.5. The LTE-AKA based part (§6.2.1) will remain unchanged, hence we do not discuss it further. The 5G-AKA based part (§6.2.2) will require some amendments.

While combining the two solutions, the overall structure and message flow of §6.2.2 remains unaffected. The only difference is in the way the SUCI is calculated, and transmission of additional parameters along with a few of the original messages. Figure 6.4 shows the overall message flow and structure of the combined solution and highlights the various steps that are different from those of §6.2.2. We further explain each of these individual steps as follows:

1. A 5G SN requests the SUPI from the UE.
2. The UE generates a SUCI. Here, the SUCI is computed according to the details provided in §5.5. However, as the downgrade protection solution also requires the two counters, δ_{min} and δ_{max}, to be sent to the HN along with the MSIN, the SUCI calculation here caters for it. Figure 6.5 details the construction of the SUCI.
3. The UE sends the SUCI to the SN.

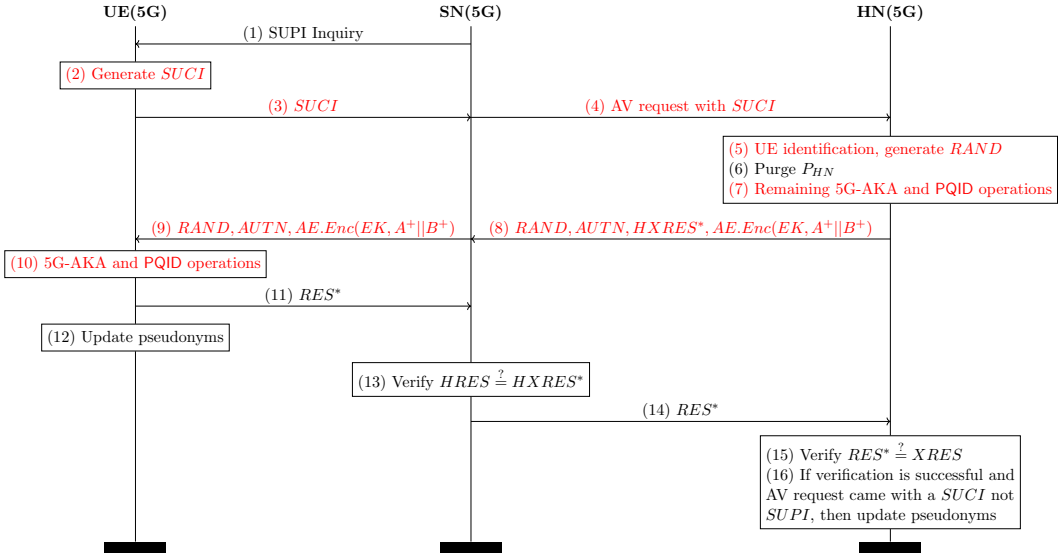


Figure 6.4: Combining PQID with 5G-AKA based downgrade protection solution. The differences to Figure 6.3 are highlighted in red.

4. The SN forwards the SUCI to the HN, requesting an AV.

5. The HN extracts $MSIN$, δ_{min} and δ_{max} from the encrypted part of the SUCI according to Figure 6.5. Further, the HN constructs the RAND by embedding a pseudonym in it as follows:

   - HN checks if p_f is NULL. If yes, an m-bit long p_f is randomly allocated (from the pool of free pseudonyms) and d_f is set to CTR, which is a subscriber-specific counter maintained by the HN. It increases every time the HN generates a new pseudonym.
   - HN checks whether δ_{max} is greater than d_f . If yes, it sets ECF to 1, otherwise ECF is set to 0.
   - A l-bit long random $salt$ is chosen.
   - $(p, d) \leftarrow (p_f, d_f)$.
   - $(p, d, ECF, salt)$ is encrypted with key k. The resultant ciphertext is $RAND$.

6. HN removes pseudonyms from P_{HN} which have counter smaller than δ_{min}.

7. HN performs other operations of 5G-AKA (except the construction of $RAND$) and PQID (as per Figure 6.5).

8. HN then sends an AV $(RAND, AUTN, HXRES^*)$ to the SN along with the encrypted update parameters $AE.Enc(EK, A^+||B^+)$.

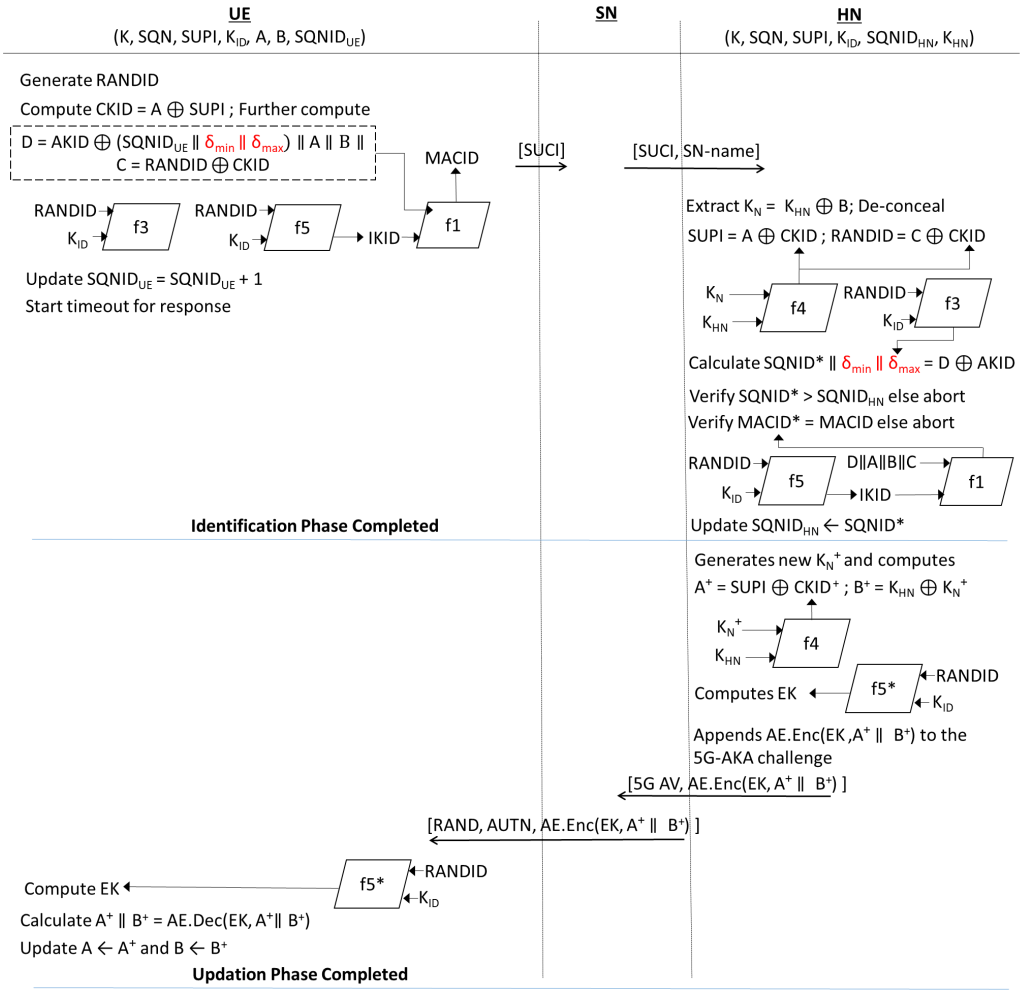


Figure 6.5: The amended PQID for the combined solution. The differences to Figure 5.3 are highlighted in red.

9. SN forwards the $RAND$, $AUTN$ and $AE.Enc(EK, A^+||B^+)$ to the UE.

10. UE performs the 5G-AKA and PQID related operations (see Figure 6.5).

11. UE then sends the response RES* to the SN.

12. UE further decrypts RAND, extracts the embedded pseudonym from the RAND, and updates the pseudonyms in the UE. In 5G-AKA the ECF might be set to 1 by the HN. In this case the UE will empty the set P_{UE}, set $(p_1, d_1), (p_2, d_2) \leftarrow (p, d-1), (p, d)$ and terminates the algorithm at this step. This is needed to recover from a very unlikely error situation where d_2 gets corrupted in the UE.

13. SN computes HRES* as a function of RES* and then compares HRES* with HXRES*.

14. If the comparison in Step (13) matches, SN forwards the RES* to the HN.

15. The HN compares RES* and XRES.

16. If the comparison in Step (15) matches, HN checks whether the AV (associated with the current 5G-AKA run) came with a SUCI or a SUPI. If with a SUCI, HN checks if the pseudonym p that was embedded in the RAND is still p_f. If yes, the HN moves (p_c, d_c) to P_{HN} and sets $(p_c, d_c), (p_n, d_n), (p_f, d_f) \leftarrow (p_n, d_n), (p_f, d_f), (NULL, NULL)$. Consequently, the HN will embed a new pseudonym in the RAND while responding to the next AV request. If the UE identified itself with GUTI or responded to a paging message, then the subsequent AV request sent by the SN will be with a SUPI. Hence, the pseudonyms will not get updated in HN. This means, in response to the next AV request, the HN will embed the same pseudonym in the RAND.

6.5 Discussion

A quick look at Figure 6.5 shows that the impact of this amalgamation is minimal upon our original PQID (§5.5) proposal. The only difference is that, in addition to masking of the MSIN, the two counters δ_{min} and δ_{max} are also masked. This can easily be accommodated by truncating the 128-bit masking parameter $AKID$ (see §5.5.2) to 96 bits instead of the 48 bits in the original scheme. Consequently, the formal security analysis of PQID (§5.7) and the results also remain valid for these changes. We therefore omit repeating the details here.

With regards to the impact of this amalgamation upon the original downgrade protection proposal of [89], as mentioned before the LTE-AKA based part remains unaffected. The changes to the 5G-AKA based part retain the original properties provisioned by the solution, as detailed in §6.4. The original message structure is maintained after the amalgamation, avoiding any additional communication overhead. Moreover, due to replacement of the public-key cryptography with symmetric cryptographic operations, a reduction in computational overhead is expected. Achieving resynchronization and provisioning of LI remains unaffected after this amalgamation because both these features are related to the LTE-AKA based part of the downgrade protection proposal, which remains unaltered.

6.6 Chapter Summary

This chapter explored the feasibility of combining our symmetric private identification scheme (PQID, for details see §5.5) with the downgrade protection proposal by Khan et al. [89] to come up with a 5G identification mechanism that is both quantum-secure and downgrade-resistant. We showed how the combining of our PQID scheme with the downgrade protection proposal can be undertaken in a seamless manner. The impact of this amalgamation is minimal upon both our PQID scheme and the downgrade protection proposal of [89].

Chapter 7

Privacy-Preserving Key Agreement for IEEE Std 802.15.6

In this chapter we propose two key agreement protocols for the international WBAN standard IEEE Std 802.15.6. In addition to the requisite security properties, the proposed protocols also offer privacy guarantees. We develop a formal security and privacy model and prove the proposed protocols secure in this model.

7.1 Introduction

The security of traffic in IEEE Std 802.15.6 is protected using authenticated encryption, which requires the establishment of symmetric keys. The procedure for agreeing these keys is thus critical to the overall security and privacy of a WBAN. As discussed in Chapter 2, the key agreement protocols of IEEE Std 802.15.6 have been shown to have security weaknesses [134]. In this chapter, we present two key agreement protocols for IEEE Std 802.15.6 which render a comprehensive range of security and privacy properties, which are regarded as essential [102] for WBANs. We start by elaborating upon the desired security, privacy and functional objectives.

7.1.1 Desired Objectives

The list of the requisite properties (and, where required, the associated rationale) of a Privacy-Preserving Key Agreement (PPKA) protocol to be executed between a node N and HN is as below:

7.1.1.1 Security Properties

- **Mutual Entity Authentication.** Entity authentication is the process by which one entity (the verifier) is assured of the identity of a second entity (the

claimant) [139]. The PPKA protocol should provision mutual entity authentication between N and HN.

- **Mutual "Implicit" Key Authentication.** The assurance that only a particularly identified other party may possibly know the negotiated key [139]. Mutual "implicit" key authentication is required between N and HN.
- **Known Key Security.** An adversary compromising a session key in a single session should not impose any threat to the session key security in any other sessions.
- **Key Randomness.** The assurance that any successful key agreement should output a uniformly distributed session key among the set of all possible session keys [133].
- **Partial Forward Secrecy.** The compromise of the long-term secret of a node N should not enable an adversary to compromise previously established session keys of that node. Partial Forward Secrecy (PrFS) is crucial as client nodes (unlike HN) in typical WBAN deployment scenarios are not tamper-proof and their internal storage can be accessed by an adversary easily. Note that, as already explained in §2.11, we do not consider the compromise of the long-term secret of HN. This enables us to consider a more pragmatic version of forward secrecy for WBANs. PrFS is a well-documented [41] and discussed [40, 46, 47, 121] security notion for key exchange protocols. In contrast to the standard notion of Perfect Forward Secrecy (PFS), PrFS considers the compromise of the long-term secret of only one of the protocol participants. We remark that PrFS is distinct from the related notion of Weak Forward Secrecy (WFS) [96], where the concerned adversary is a passive one. PrFS considers an active adversary.
- **Key Compromise Impersonation (KCI) Resilience.** Suppose N's long-term secret gets disclosed. Clearly an adversary that knows this value can now impersonate N, since it is precisely this value that identifies N. However, it is highly desirable that this loss should not enable an adversary to impersonate other entities to N [100]. Consider the scenario where a cardiac pacemaker is part of a WBAN deployed upon a chronic patient by a hospital for remote administration and monitoring purposes. The leakage of the pacemaker's long-term secret should not enable the adversary to issue "stop" commands to the pacemaker by impersonating as the hospital administrator. Such a case could potentially lead to a life-threatening situation.

- **Replay Prevention.** An adversary should not be able to successfully replay previously captured copies of legitimate messages between the protocol participants.
- **Desynchronization Resistance.** If the authentication parameters get updated during the protocol execution, then usually the participants need to have the same updated values at the end of a protocol run. Otherwise, they will not authenticate each other in later associations and we say they have been desynchronized. In a desynchronization attack, the adversary forces the protocol participants to update their authentication parameters to different values. A PPKA needs to be resistant to these types of attacks.

7.1.1.2 Privacy Properties

We focus on two privacy aspects:

1. **Node Anonymity.** An adversary $\mathcal{A}$, who is observing all communications, should not be able to learn the identity of any node N who is participating in a PPKA protocol with HN. The privacy attribute of anonymity is a necessity for typical application scenarios of WBANs, such as healthcare and military.
2. **Unlinkability.** An adversary $\mathcal{A}$, who is observing communications, should not be able to link one successfully-executed PPKA instance of node N to another successfully-completed instance of the same node. Unlinkability is imperative in addition to anonymity. Although the PPKA instances could be anonymous, if the adversary is able to link various PPKA instancess and group them together then $\mathcal{A}$ would be able to attribute a group of captured instances to a particular node with high probability, due to his knowledge of the operations of the WBAN. For example, consider a medical WBAN in which a pacemaker is to communicate with the remote healthcare providers every five minutes, while the body temperature sensor communicates only three times per day.

7.1.1.3 Functional Requirements

- **Support for Multi-Hop Communication.** As discussed in §2.11, depending upon the network topology, nodes would either be communicating directly

with the Hub Node HN or via an Intermediary Node IN. Therefore, the PPKA protocol should be designed to be suitable for both single-hop and two-hop communication modes of [3].

- **Energy Consumption.** As nodes in a WBAN are severely energy-constrained, the PPKA protocol needs to be minimalistic in terms of computation, communication and storage overhead. Energy consumption in WBANs is dominated by radio communications [54], which mainly depends on the number of bits to be transmitted within the network. Consequently, the PPKA protocol should be designed such that the number of bits to be exchanged between the protocol participants and the computational overhead for nodes N should be minimal.
- **Stateless HN.** HN is the consistent nucleus of the network whose lack of accessibility will have devastating effects on the complete WBAN. As the network topology in WBANs is dynamic (client nodes join and leave the network on a frequent basis), it is imperative for HN's accessibility that it be independent of such dynamism. Consequently, an important requirement is that the PPKA protocol should not require HN to maintain a state of the WBAN nodes.

7.1.2 Design Principles

During the design of the PPKA protocols for IEEE Std 802.15.6 we focus on the following principles:

- **Offloading of Expensive Operations.** As nodes in a WBAN are resource-constrained, it makes sense to offload energy-expensive operations to more resourceful entities such as SA and HN. An example of this is discussed in more detail in §7.4.2.
- **Minimizing the Implementation Footprint.** Ideally, the proposed solution should not introduce new cryptographic primitives as this will adversely affect the implementation footprint (both hardware and memory). Specifically, we aim to use the already specified block cipher function in [3] for achieving the various security and privacy objectives. A more detailed discussion is given in §7.4.2.
- **Reducing Management Costs.** A PPKA solution should not place management costs on the WBAN nodes after the network initialization. Consider

the situation where a third party wants to add its node (for example a fitness tracker) to an already deployed WBAN. The third party should be able to contact the SA who (after registration of the new node) dispatches it to the WBAN owner, who begins using the new device upon receipt. Note that all this was done without any interaction between the SA and the currently operational WBAN.

7.1.3 Related Work

Toorani [134] discovered various security weaknesses in the key agreement methods of IEEE Std 802.15.6, all of which were susceptible to KCI attacks (see §7.1.1.1), as well as attacks on forward secrecy. Wang and Zhang [142] proposed a key agreement scheme for WBANs that claimed to provide anonymity and unlinkability in addition to the requisite security guarantees. However, Jiang et al. [76] showed that the scheme in [142] is vulnerable to a client impersonation attack. They proposed an authenticated key agreement scheme which rectified this flaw. However, their scheme was based on computing bilinear pairings [64], which means that it is not suitable for deployment in resource-constrained WBANs. To avoid the overhead of managing public-key certificates, He et al. proposed a certificateless authentication scheme [71], which provides anonymity and unlinkability. However, the computation and communication overheads associated with their scheme also render it unsuitable for WBAN deployment. Li et al. [103] presented an authenticated key agreement scheme based only upon symmetric cryptographic primitives. This is an attractive proposal since there is no requirement of any additional infrastructure and the associated computation and communication overheads are negligible. The authors claimed that this scheme achieved almost all of the security and privacy objectives defined in §7.1.1.

7.1.4 Contributions

The main contributions of this chapter are as follows:

- We provide an analysis of [103] which, in addition to showing that Li et al.'s scheme does not provide unlinkability and forward secrecy, also exhibits its vulnerability to KCI attacks.
- We propose two key agreement protocols (PPKA-1 and PPKA-2) which provide

Table 7.1: Comparison of security and privacy features.

Security/Privacy Feature	**Li et al. [103]**	PPKA-1 [§7.3.1]	PPKA-2 [§7.3.2]
PrFS	✗	✗	✓
KCI Resilience	✗	✗	✓
Unlinkability	✗	✓	✓
Anonymity	✓	✓	✓

unlinkability and resolve the privacy flaws found in [103]. PPKA-2 additionally provisions PrFS and KCI resilience. Table 7.1 lists the security and privacy features provisioned by each protocol.

- We develop a formal security and privacy model in an appropriate complexity-theoretic framework and prove the proposed protocols secure in this model.

The rest of this chapter is organized as follows: §7.2 details Li et al.'s scheme while §7.3 presents our PPKA protocols. §7.4 discusses the design decisions regarding the PPKA protocols. §7.5 presents the formal security model followed by §7.6, which analyzes the PPKA protocols. Finally, §7.7 concludes the chapter.

7.2 Li et al.'s Scheme

In this section we present an overview and analysis of Li et al.'s scheme [103]. For ease of comparison we use the same notation (details in Table 7.2) as used in [103].

7.2.1 The Key Agreement Protocol

Li et al.'s PPKA protocol between the HN and N consists of three phases. For a pictorial overview of the protocol see Figure 7.1.

7.2.1.1 Initialization Phase

The SA generates a master secret key k_{HN} and stores it in HN.

Table 7.2: Notations used in Li et al.'s protocol.

Symbol	Description
$h(.)$	Cryptographic hash function
(a, b)	Concatenation of a and b
$\oplus$	Bitwise XOR operation k
SA	System Administrator (initializes the WBAN)
N	Normal Node
HN	Hub Node
IN	Intermediary Node
id_N	Long term secret/identity of node N
id'_{IN}	Relay identity of node IN
tid_N	Temporary identity of node N
k_{HN}	Long term master secret key of HN
k_N, f_N	Temporary secret parameters chosen by HN/SA
r_N	Temporary secret parameter chosen by N
a_N, b_N	Authentication parameters stored in N
x_N, y_N	Auxiliary authentication parameters
α, β, η, μ	Authentication parameters computed by HN
k_S	Resultant shared key
t_N	Timestamp generated by node N
$X \rightarrow Y : Z$	Entity X sends message Z to entity Y

7.2.1.2 Registration Phase

The SA generates a unique secret identity id_N for node N. It then randomly chooses the temporary secret parameter k_N and calculates $a_N = id_N \oplus h(k_{HN}, k_N)$ and $b_N = k_{HN} \oplus a_N \oplus k_N$. A unique relay identity id'_{IN} for the intermediary node IN is chosen and the parameters $\langle id_N, a_N, b_N \rangle$ and $\langle id'_{IN} \rangle$ are stored in N and IN respectively, while id'_{IN} is stored by HN as the identity of IN when communicating in relay mode.

7.2.1.3 Authentication Phase

We can think of the authentication phase of Li et al.'s scheme as a two-pass protocol. The individual steps are outlined below:

Step 1: $N \rightarrow IN : \langle tid_N, y_N, a_N, b_N, t_N \rangle$. N picks a random r_N and creates timestamp t_N. Then it computes $x_N = a_N \oplus id_N$, $y_N = x_N \oplus r_N$ and $tid_N = h(id_N \oplus t_N, r_N)$ and forwards the tuple $\langle tid_N, y_N, a_N, b_N, t_N \rangle$ to IN.

Step 2: $IN \rightarrow HN : \langle tid_N, y_N, a_N, b_N, t_N, id'_{IN} \rangle$. IN adds its relay identity id'_{IN} to

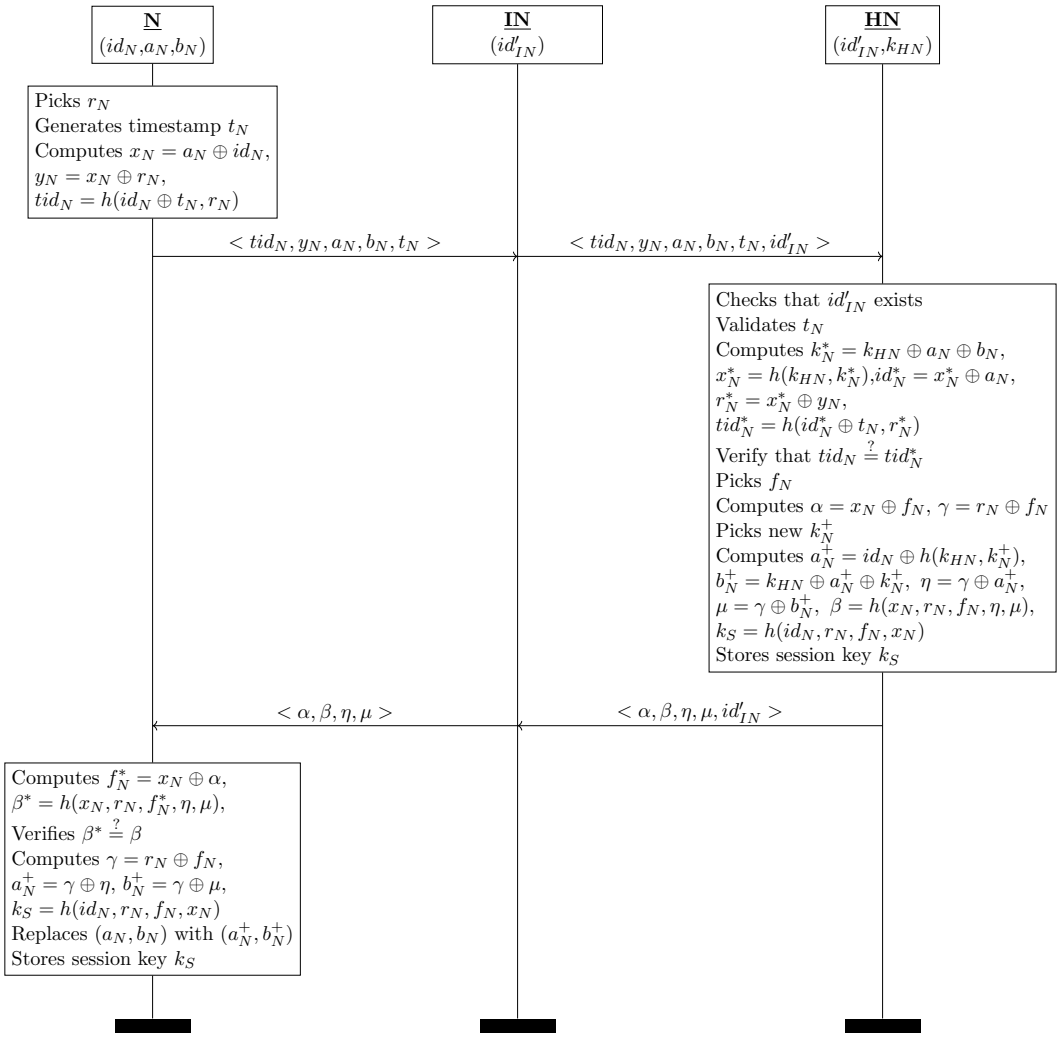


Figure 7.1: Li et al.'s protocol.

the tuple and forwards it to HN. Note that IN, when operating in relay mode, uses id'_{IN} not id_{IN}.

Step 3: $HN \rightarrow IN : \langle \alpha, \beta, \eta, \mu, id'_{IN} \rangle$. After receiving the parameters from IN, HN verifies the relay identity id'_{IN} from its database and substantiates the validity of the timestamp t_N. Upon success of these checks, it computes $k_N^* = k_{HN} \oplus a_N \oplus b_N$, $x_N^* = h(k_{HN}, k_N^*)$, $id_N^* = x_N^* \oplus a_N$, $r_N^* = x_N^* \oplus y_N$ and $tid_N^* = h(id_N^* \oplus t_N, r_N^*)$. It then verifies whether $tid_N \stackrel{?}{=} tid_N^*$. Then, a random f_N is chosen and $\alpha = x_N \oplus f_N$ and $\gamma = r_N \oplus f_N$ are computed. A new k_N^+ is picked and $a_N^+ = id_N \oplus h(k_{HN}, k_N^+)$, $b_N^+ = k_{HN} \oplus a_N^+ \oplus k_N^+$, $\eta = \gamma \oplus a_N^+$, $\mu = \gamma \oplus b_N^+$, $\beta = h(x_N, r_N, f_N, \eta, \mu)$ are computed. The shared key is computed as $k_S = h(id_N, r_N, f_N, x_N)$ and is stored in memory. Finally, HN forwards the tuple $\langle \alpha, \beta, \eta, \mu, id'_{IN} \rangle$ to IN.

Step 4: $IN \rightarrow N : \langle \alpha, \beta, \eta, \mu \rangle$. IN removes the relay identity id'_{IN} from the received tuple and forwards $\langle \alpha, \beta, \eta, \mu \rangle$ to N.

Step 5: Upon receipt of the response from IN, N computes $f_N^* = x_N \oplus \alpha$ and $\beta^* = h(x_N, r_N, f_N^*, \eta, \mu)$ and verifies that $\beta^* \stackrel{?}{=} \beta$. If true, N computes $\gamma = r_N \oplus f_N$, $a_N^+ = \gamma \oplus \eta$ and $b_N^+ = \gamma \oplus \mu$. The shared key k_S is computed as $h(id_N, r_N, f_N, x_N)$ and the authentication parameters (a_N, b_N) are replaced by (a_N^+, b_N^+).

7.2.2 Analysis of the Li et al.'s Scheme

In this section we discuss vulnerabilities and attacks on the security and privacy of Li et al.'s scheme.

7.2.2.1 Security Analysis

In addition to provisioning of mutual authentication [55], Li et al.'s scheme fulfills all the security criteria as defined in §7.1.1 except KCI resilience and PrFS. Moreover, the scheme also protects the master secret (k_{HN}) in the event of compromise of various nodes of the WBAN. For sake of brevity, we will restrict our security analysis to highlight only the vulnerabilities of Li et al.'s scheme.

Discussion about Forward Secrecy. Li et al. claimed a forward security property of their scheme. Their definition of forward secrecy varies from the generally accepted one. According to Li et al., the goal of forward secrecy is to protect other (past/future) keys in the event of compromise of the current key k_S. However, the conventional definition of forward secrecy states that in the event of compromise of the long-term secrets of the protocol participant(s), an adversary should not be able to obtain any of the past keys [107]. While Li et al.'s scheme is forward secure according to their own definition, it is not forward secure in a conventional sense.

KCI Attack. We demonstrate a KCI attack on Li et al.'s scheme. $\mathcal{A}$ observes the first pass of the protocol and notes the message contents. As the value id_N is known to $\mathcal{A}$, he calculates the following values as follows:

$$x_N = a_N \oplus id_N; \quad r_N = y_N \oplus x_N.$$

$\mathcal{A}$ chooses a random f_N and calculates $\alpha = f_N \oplus x_N$. $\mathcal{A}$ then chooses arbitrary values of η and μ and calculates β as:

$$\beta = h(x_N, r_N, f_N, \eta, \mu).$$

Finally, $\mathcal{A}$ sends the tuple $\langle \alpha, \beta, \eta, \mu \rangle$ back to node N. N cannot detect this KCI attack as N's computed value β is the same as in the received tuple. As a result, node N will end up sharing the key $k_S = h(id_N, r_N, f_N, x_N)$ with $\mathcal{A}$, incorrectly believing itself to be sharing k_S with HN.

7.2.2.2 Privacy Analysis

The Anonymity Dilemma. It is known *a priori* to the attacker that all nodes ultimately communicate with HN. As the node identifier id_N is always masked (by taking an XOR of it with a fresh random value), anonymity in Li et al.'s protocol is preserved from "direct" privacy attacks. However, now consider the situation depicted in Figure 7.2, where an intermediary node IN is providing the relaying service to various nodes. In the second pass of Li et al.'s scheme, it is not clear

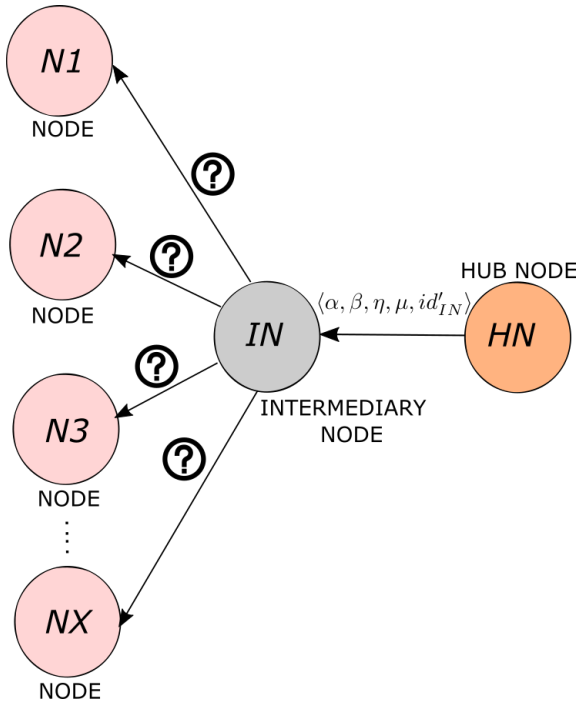


Figure 7.2: The privacy dilemma of Li et al.'s scheme.

how the intermediary node IN would be able to identify the original node N out of the "anonymity set" [122] for onward forwarding of the tuple $\langle \alpha, \beta, \eta, \mu \rangle$ received from HN. One naive way to resolve this is to allow IN to broadcast the second pass

of protocol for all nodes. However, this approach is unsuitable for already energy-constrained WBAN nodes as they will need to perform additional communication (radio reception) and computational steps for each transmission.

Unlinkability. While Li et al. claim their scheme provides unlinkability, we show this to be untrue. We highlight a weakness in Li et al.'s key agreement protocol, which allows a passive attacker to easily link two or more key agreement instances of the same node N. The attack proceeds as follows:
Instance # 1. Suppose that a run of Li et al.'s key agreement protocol is being carried out between node N and HN. A passive attacker $\mathcal{A}$ observes the contents of the messages being exchanged. From Step 1 of §7.2.1.3, $\mathcal{A}$ records the value $y_N = x_N \oplus r_N$. Then, from Step 3 of §7.2.1.3, $\mathcal{A}$ records $\alpha = x_N \oplus f_N$. Now, $\mathcal{A}$ obtains the value $\gamma = r_N \oplus f_N = \alpha \oplus y_N$. Further, $\mathcal{A}$ records the values η and μ from Step 3 of §7.2.1.3 and uses γ to compute:

$$a_N^+ = \gamma \oplus \eta; \quad b_N^+ = \gamma \oplus \mu.$$

Instance # 2. Now, $\mathcal{A}$ observes key exchange protocol exchanges between various nodes and HN. $\mathcal{A}$ compares the values of the parameters a_N and b_N from Step 1 of the protocol with the saved values of a_N^+ and b_N^+. When $\mathcal{A}$ finds a match, $\mathcal{A}$ concludes with almost certainty that another key exchange instance has been initiated by the same node N. This is correct because node N uses the updated authentication parameters a_N^+ and b_N^+ in its next run of the protocol. In this way, $\mathcal{A}$ can track and link instances of node N, demonstrating that Li et al.'s scheme does not achieve unlinkability.

7.2.2.3 Functional Requirements

Li et al.'s scheme can easily be adapted for direct communication between N and HN without the involvement of IN. Since this scheme employs only symmetric cryptographic primitives, it is extremely efficient from a computation, communication and storage-overhead perspective and there is no requirement for any additional network infrastructure. Assuming a hash function with a digest length of B-bits and 16-bit intermediary node IDs (i.e. id'_{IN}), Table 7.3 highlights the communication, computation and storage overhead of Li et al.'s scheme. In this table, h denotes one hash operation, $\oplus$ denotes an XOR operation and m denotes the number of intermediary nodes in the WBAN.

Table 7.3: Overheads associated with Li et al.'s scheme.

Index	Node N	Hub Node HN
Computation Overhead	$3h + 7\oplus$	$5h + 12\oplus$
Communication Overhead	$5B$ bits	$4B + 16$ bits
Storage Overhead	$3B$ bits	$(B + 16m)$ bits

Table 7.4: Detail of additional symbols.

Symbol	Description
id'_N	Temporary identity chosen randomly by N
z_N	Security parameter stored in memory of N by HN/SA
$\mathsf{Enc}(k, m)$	Encryption of message m under symmetric key k
$\mathsf{Dec}(k, c)$	Decryption of ciphertext c under symmetric key k
γ	Additional authentication parameter computed by HN

Note that, contrary to the assumption made by Li et al. in §5.4 of [103] about the arbitrary length of the timestamp field, it is implicitly the same length as the hash function digest because, as described earlier in §7.2.1.3, $tid_N = h(id_N \oplus t_N, r_N)$. This is not commensurate with the length of the timestamp field as defined in IEEE Std 802.15.6, which is three octets or 24-bits. Regarding state maintenance by HN, in case of [103], HN needs to maintain states concerning the relay nodes IN, which is an undesirable feature as already explained in §7.1.1.3.

7.3 Our PPKA Protocols

In this section we propose two PPKA protocols which rectify the problems highlighted in §7.2.2. While devising these PPKA protocols, we have tried to preserve the original elegance, simplicity and efficiency of the scheme in [103]. The first PPKA protocol (PPKA-1) addresses the privacy flaws of unlinkability and anonymity dilemma faced by IN (§7.2.2.2) in Li et al.'s scheme. The second protocol (PPKA-2), additionally provides PrFS and KCI resilience (in case of compromise of the long-term secret of node N). Note that although in our protocols the intermediary node IN is not an active participant from a cryptographic standpoint (this was a conscious design consideration), we have included IN in our protocol description for verification of the resolution of the anonymity dilemma of IN. Detail of additional notation used in our PPKA protocols is given in Table 7.4.

7.3.1 PPKA-1

PPKA-1 is separated into three distinct phases:

1. An *Initialization Phase*, that generates the long-term secret values for HN.
2. A *Registration Phase*, that generates the long-term values for the rest of the nodes and stores them with HN.
3. An *Authentication Phase* where the nodes N and HN generate an authenticated shared secret key, and update the authentication parameters.

We next describe these three phases.

7.3.1.1 Initialization Phase

This phase is identical to the *Initialization Phase* of [103]. Specifically, the SA generates a master secret key k_{HN} and stores it in HN.

7.3.1.2 Registration Phase

This phase is also similar to that of [103]. However, in the case of PPKA-1, the intermediary node IN is not provided with a relay identity id'_{IN}.

7.3.1.3 Authentication Phase

The various steps of the *Authentication Phase* of PPKA-1 are depicted in Figure 7.3 and are as follows:

Step 1: $N \rightarrow IN : \langle tid_N, y_N, a_N, b_N, t_N, id'_N \rangle$. N picks a random r_N and creates timestamp t_N. It then computes $x_N = a_N \oplus id_N$, $y_N = x_N \oplus r_N$. It further picks a random pseudonym id'_N to be used as a temporary identifier for this key agreement instance only, calculates $tid_N = h(id_N, id'_N, t_N, r_N)$ and sets the "Relay Field" of the underlying "MAC Header" to value 1, according to sub-clause 6.10 of [3].

Step 2: $IN \rightarrow HN : \langle tid_N, y_N, a_N, b_N, t_N, id'_N \rangle$. IN checks the value of "Relay Field" and forwards the tuple to HN.

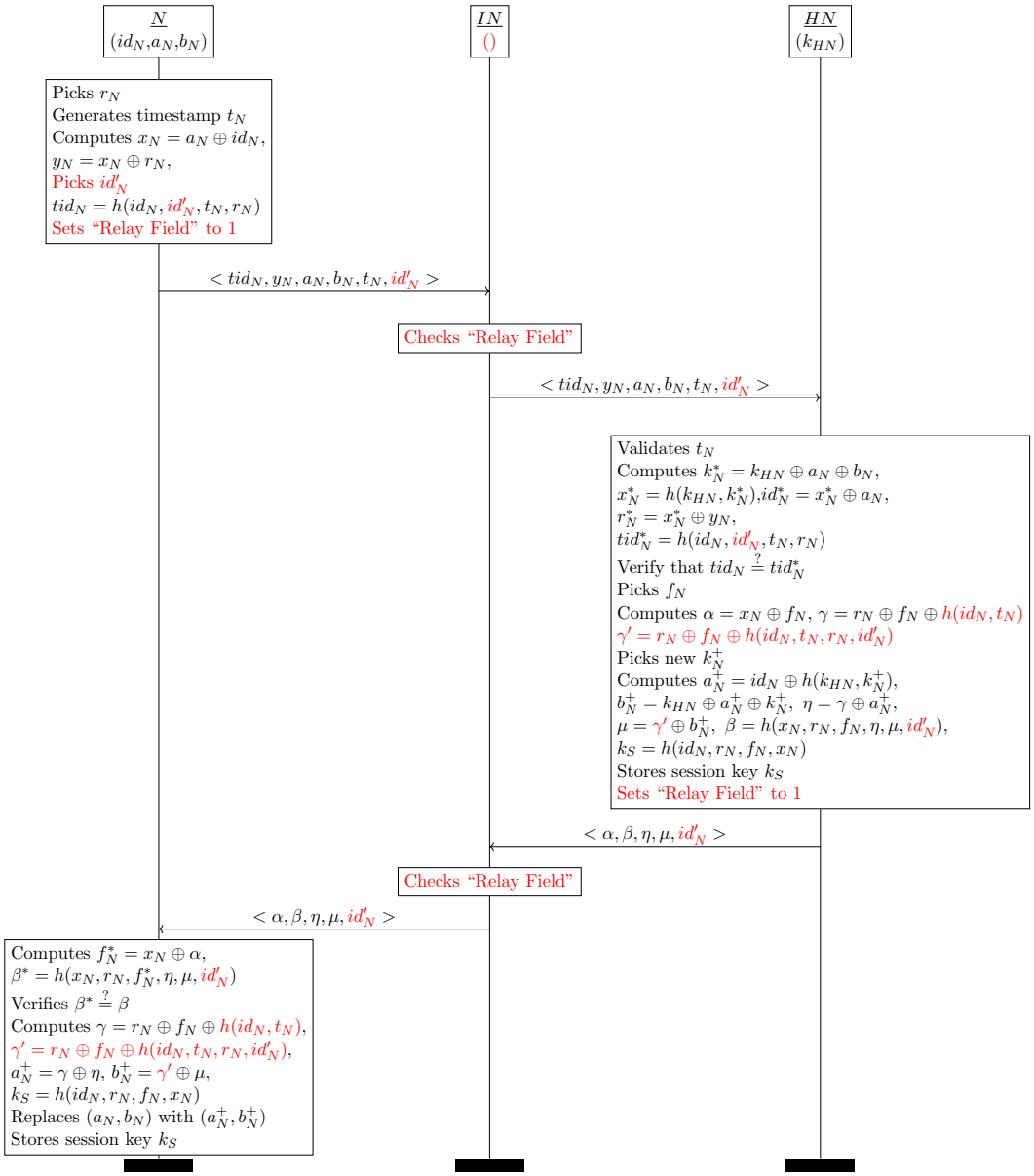


Figure 7.3: Protocol PPKA-1. Steps different from Li et al.'s protocol (Figure 7.1) are highlighted in red.

Step 3: $HN \rightarrow IN : \langle \alpha, \beta, \eta, \mu, id'_N \rangle$. After receipt of the tuple from IN, HN verifies the validity of the timestamp t_N. Upon success of this check, it computes $k^*_N = k_{HN} \oplus a_N \oplus b_N$, $x^*_N = h(k_{HN}, k^*_N)$, $id^*_N = x^*_N \oplus a_N$, $r^*_N = x^*_N \oplus y_N$ and $tid^*_N = h(id^*_N, id'_N, t_N, r^*_N)$. It then verifies whether $tid_N \stackrel{?}{=} tid^*_N$. Then, a random f_N is chosen and $\alpha = x_N \oplus f_N$, $\gamma = r_N \oplus f_N \oplus h(id_N, t_N)$ and $\gamma' = r_N \oplus f_N \oplus h(id_N, t_N, r_N, id'_N)$ are computed. Then a new k^+_N is picked and $a^+_N = id_N \oplus h(k_{HN}, k^+_N)$, $b^+_N = k_{HN} \oplus a^+_N \oplus k^+_N$, $\eta = \gamma \oplus a^+_N$, $\mu = \gamma' \oplus b^+_N$, $\beta = h(x_N, r_N, f_N, \eta, \mu, id'_N)$ are computed. Finally, the shared key $k_S = h(id_N, r_N, f_N, x_N)$ is computed and stored in memory,

and the value of the underlying "Relay Field" is set to 1.

Step 4: $IN \rightarrow N : \langle \alpha, \beta, \eta, \mu, id'_N \rangle$. IN checks the "Relay Field" of the message received from HN. If "Relay Field" value is set to 1, then it notes the identifier id'_N received in the tuple for onward forwarding of the tuple to node N.

Step 5: Upon receiving a response from IN, N computes $f^*_N = x_N \oplus \alpha$ and $\beta^* = h(x_N, r_N, f^*_N, \eta, \mu, id'_N)$ and verifies that $\beta^* \stackrel{?}{=} \beta$. If so, N computes $\gamma = r_N \oplus f_N \oplus h(id_N, t_N)$, $\gamma' = r_N \oplus f_N \oplus h(id_N, t_N, r_N, id'_N)$, $a^+_N = \gamma \oplus \eta$ and $b^+_N = \gamma' \oplus \mu$. The shared key k_S is computed as $h(id_N, r_N, f_N, x_N)$, and the authentication parameters (a_N, b_N) are updated by being replaced with (a^+_N, b^+_N).

7.3.2 PPKA-2

The second PPKA protocol PPKA-2 is structurally similar to PPKA-1. We now describe the execution of PPKA-2.

7.3.2.1 Initialization Phase

This phase is unchanged from PPKA-1.

7.3.2.2 Registration Phase

The *Registration Phase* is also identical to PPKA-1. However, SA additionally computes $z_N = h(k_{HN}, id_N, k_N)$. Parameters $\langle id_N, a_N, b_N, z_N \rangle$ are stored in N.

7.3.2.3 Authentication Phase

The authentication phase of PPKA-2 is depicted in Figure 7.4 and detailed as follows:

Step 1: $N \rightarrow IN : \langle tid_N, y_N, a_N, b_N, t_N, id'_N \rangle$. This is identical to Step 1 of PPKA-1 except that the value of tid_N is calculated as $h(id_N, id'_N, z_N, t_N, r_N)$.

Step 2: $IN \rightarrow HN : \langle tid_N, y_N, a_N, b_N, t_N, id'_N \rangle$. This is identical to Step 2 of PPKA-1.

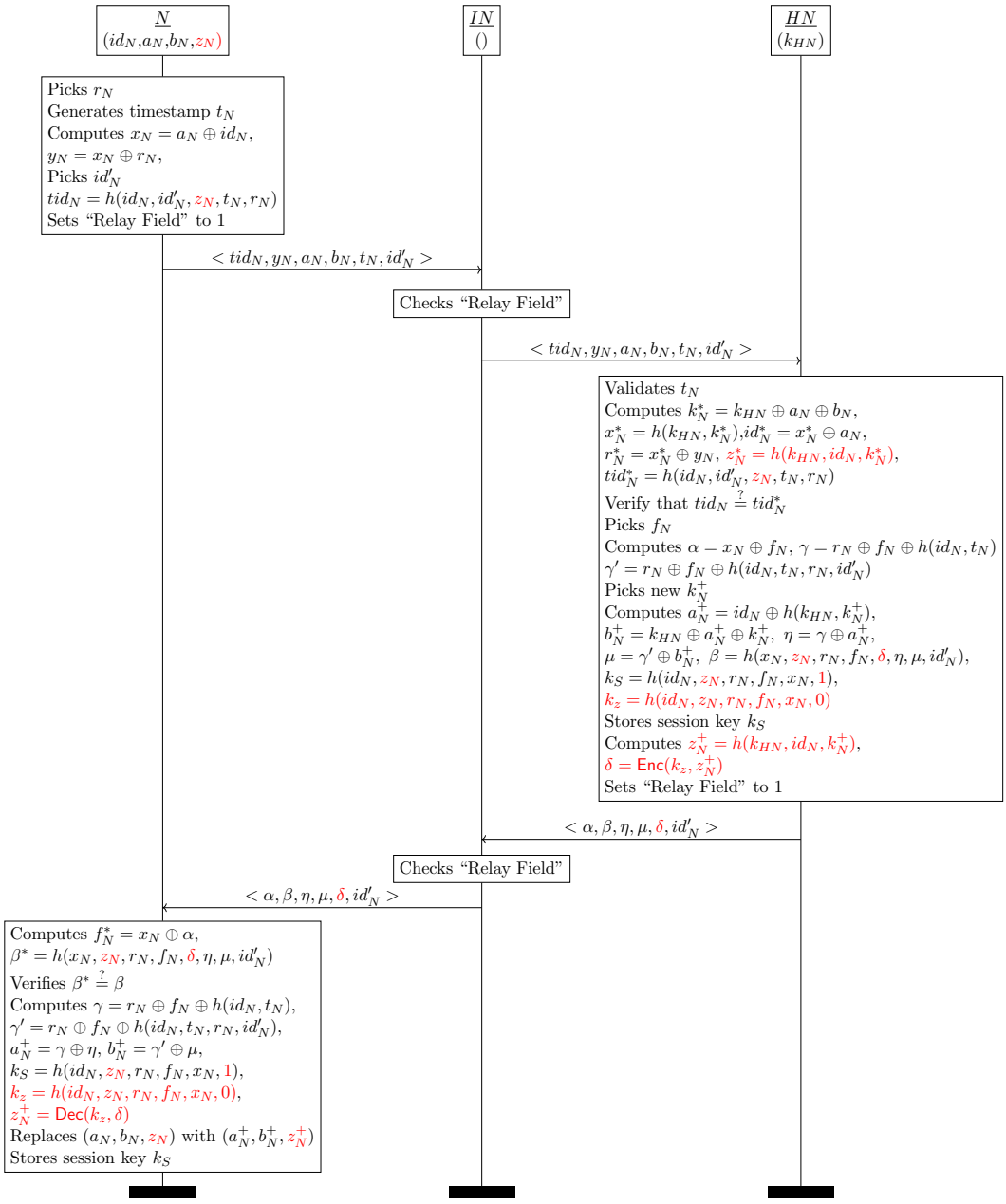


Figure 7.4: Protocol PPKA-2. Steps different from PPKA-1 (Figure 7.3) are highlighted in red.

Step 3: $HN \rightarrow IN : \langle \alpha, \beta, \eta, \mu, \delta, id'_N \rangle$. After receipt of the tuple from IN, HN proceeds identically to Step 3 of PPKA-1. Additionally z^*_N is calculated as $h(k_{HN}, id_N, k_N)$ and tid^*_N as $h(id^*_N, id'_N, z^*_N, t_N, r^*_N)$. It then verifies whether $tid_N \stackrel{?}{=} tid^*_N$. Then, α, η and μ are computed as in PPKA-1. k_S is computed as $h(id_N, z_N, r_N, f_N, x_N, 1)$ while an additional key k_Z is computed as $h(z_N, id_N, r_N, f_N, x_N, 0)$. HN then computes $z^+_N = h(k_{HN}, id_N, k^+_N)$ and encrypt it with k_z as $\delta = \mathsf{Enc}(k_z, z^+_N)$. Lastly, β is calculated as $h(x_N, z_N, r_N, f_N, \delta, \eta, \mu, id'_N)$.

Step 4: $IN \rightarrow N : \langle \alpha, \beta, \eta, \mu, \delta, id'_N \rangle$. Identical to Step 4 of PPKA-1.

Step 5: This is identical to Step 5 of PPKA-1, except that β^* is calculated as $h(x_N, z_N, r_N, f^*_N, \delta, \eta, \mu, id'_N)$ and the shared key k_S is computed as $h(id_N, z_N, r_N, f_N, x_N, 1)$. Additionally, N decrypts $z^+_N = \mathsf{Dec}(k_z, \delta)$ and replaces z_N with z^+_N.

7.4 Discussion

7.4.1 Why a Bespoke Solution?

If we consider the scenario of direct communication between N and HN (without the involvement of IN), at first glance it seems to be similar to that of RFID, where a *tag* needs to be authenticated in a secure and private manner by the *reader*. However, there is a fundamental distinction between the two scenarios. As discussed in §7.1.1.3, in the WBAN case the HN does not maintain any state about the network nodes and is oblivious to the identity management of the network while, in the RFID setting, the *reader* has access to the back-end database server(s) which maintain nodes' status in the RFID network. This means that in the case of RFID, SA needs to update the status at the back-end servers whenever it introduces a new node or removes an old one from the system. As explained in §7.1.1.3, this is problematic for WBANs.

7.4.2 Random Number Generation on WBAN Nodes

The Pseudo Random Number Generator (PRNG) for a WBAN needs to be computationally inexpensive and there should be no requirement for entropy collection from environmental resources, as this would entail extra communication. To achieve this, we recommend the approach outlined in [99]. During the "Registration Phase", SA can allocate each node N with a unique (randomly chosen) secret key K. Thereafter, N can encrypt the sequence $\{0, 1, 2, 3, ...\}$ under key K using AES (already available for message security purposes) as the block cipher. This arrangement can securely generate 2^{60} bytes without the need for re-seeding the key K.

7.4.3 Post-Quantum Significance

Given recent progress towards achieving practical universal quantum computers [45], it is imperative that proposals for any standard should also cater for this future threat. Our PPKA protocols avoid any public-key cryptography and are thus well suited to post-quantum deployment scenarios.

7.4.4 Why Timestamps?

Timestamps are generally avoided in key agreement protocols as they present various practical problems, such as the need for a reliable source of time. Our setting does not face these difficulties, as a comprehensive mechanism already exists in Clause 6.11 of [3] which provisions for HN to act as the central time source for the WBAN and regularly broadcasts time-synchronization beacons.

7.4.5 Why Two Proposals?

As already highlighted in Table 7.1, PPKA-2 offers additional security and privacy features over and above those offered by PPKA-1. The question then arises that why there is a need for two separate proposals for IEEE 802.15.6? This is because these additional features offered by PPKA-2 come at a cost of additional computational, communication and storage overheads, the detail of which is further elaborated in Tables 7.5 and 7.6. As discussed in Section 7.1.1.3, nodes in a WBAN are severely energy-constrained; hence, the PPKA protocol needs to be minimalistic in terms of computation, communication and storage overhead. By providing two separate proposals, IEEE will have the flexibility to choose from two options that offer a trade-off between overheads and the provided features.

7.5 Security Model

We now introduce our security model for the analysis of PPKA protocols. Our first security experiment is based on standard key-exchange models in the tradition of Bellare-Rogaway [34] key indistinguishability games. This allows our model to easily capture known key secrecy, as well as generically capture key randomness notions, since our adversary is tasked merely with the goal of distinguishing the targeted session key from a random session key from the same distribution.

Our second security experiment allows us to capture privacy notions of sessions by challenging an adversary to determine which of two previously selected nodes ran a given protocol execution. Our cleanness predicates (see §7.5.4) allows us to model KCI attacks by allowing the adversary to reveal the long-term key of the node running the PPKA protocol, as well as the notions of partial forward secrecy. We begin by describing the execution environment for our security frameworks.

7.5.1 Execution Environment

Consider an experiment $\mathsf{Exp}^{\mathsf{PPKA\text{-}IND}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$ played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. $\mathcal{C}$ maintains a single node HN, running a number of instances of the PPKA protocol Π, and a set of (up to) n_N nodes $N_1, \ldots, N_{n_N}$ (representing nodes communicating with the hub node HN), each potentially running one stage of (up to) n_S consecutive stages of Π. The PPKA protocol Π is represented as a tuple of algorithms $\Pi = (\mathsf{HKeyGen}, \mathsf{HF}, \mathsf{NKeyGen}, \mathsf{NF}, \mathsf{StateGen}, \mathsf{StateUpdate})$. We abuse notation and use π_i^s to refer to both the identifier of the $stid$-th stage of Π being run by node N_{id} and the collection of per-session variables maintained for this stage. We describe the algorithms as follows:

- $\Pi.\mathsf{HKeyGen}(\lambda) \xrightarrow{\$} (k_{HN})$ is a probabilistic symmetric key generation algorithm taking as input a security parameter λ and outputting a long-term hub node secret key (k_{HN}).
- $\Pi.\mathsf{HF}(\lambda, k_{HN}, m) \xrightarrow{\$} (m')$ is a (potentially) probabilistic algorithm that takes a security parameter λ, the long-term key of the hub node k_{HN}, and an arbitrary bit string $m \in \{0,1\}^* \cup \{\emptyset\}$, and outputs a response $m' \in \{0,1\}^* \cup \{\emptyset\}$ and an updated per-session state π'.
- $\Pi.\mathsf{NKeyGen}(\lambda) \xrightarrow{\$} (ltk)$ is a probabilistic symmetric key generation algorithm taking as input a security parameter λ and outputting a long-term hub node secret key (ltk). Note that in our proposed PPKA protocols, we denote this long-term secret key with id_N.
- $\Pi.\mathsf{NF}(\lambda, \pi, m) \xrightarrow{\$} (m', \pi')$ is a probabilistic algorithm taking a security parameter λ, the set of per-session variables π and an arbitrary bit string $m \in \{0,1\}^* \cup \{\emptyset\}$, and outputs a response $m' \in \{0,1\}^* \cup \{\emptyset\}$ and an updated per-session state π'.

- $\Pi.\mathsf{StateGen}(\lambda, k_{HN}, ltk) \xrightarrow{\$} (psstate)$ is a probabilistic symmetric key generation algorithm taking as input a security parameter λ and the long-term secret keys of the hub node and the "normal" node, outputting secret state information for node N ($psstate$). In PPKA-1, this per-stage secret state is $\langle a_N, b_N \rangle$. In PPKA-2, this is $\langle a_N, b_N, z_N \rangle$.
- $\Pi.\mathsf{StateUpdate}(\lambda, \pi) \xrightarrow{\$} (psstate)$ is a probabilistic symmetric key generation algorithm taking as input a security parameter λ and a set of per-session variables, outputting the next stage's per-stage secret state ($psstate$) for node N.

The experiment begins with $\mathcal{C}$ running $\Pi.\mathsf{HKeyGen}$ once to generate a long-term secret key for the hub node (k_{HN}), and randomly sampling a bit $b \in \{0,1\}$. $\mathcal{A}$ then interacts with $\mathcal{C}$ via the queries listed in §7.5.2, eventually terminating and outputting a guess bit b' of $\mathcal{C}$'s bit b. $\mathcal{A}$ wins the key-indistinguishability game if $b' = b$ and the session π_i^s, such that $\mathcal{A}$ issued $\mathbf{Test}(id, stid)$, satisfies the cleanness predicate clean, which we discuss in §7.5.4. Each session maintains the following set of per-session variables:

- $ltk \in \{0,1\}^\lambda$ - the long-term symmetric-secret of N_{id};
- $id \in \{1, \ldots, n_N\}$ - the index of the node N_{id};
- $m_s \in \{0,1\}^* \cup \{\perp\}$ - the concatenation of messages sent by the node, initialised by $\perp$;
- $m_r \in \{0,1\}^* \cup \{\perp\}$ - the concatenation of messages received by the node, initialised by $\perp$;
- $psstate \in \{0,1\}^* \cup \{\perp\}$ - the per-stage secret state of the node, initialised by $\perp$;
- $sk \in \{0,1\}^* \cup \{\perp\}$ - the session key, initialised by $\perp$;
- $stid \in \{1, \ldots, n_S\}$ - the index of the most recently completed stage, initialised by 1 and increased monotonically;
- $\alpha \in \{\texttt{active}, \texttt{accept}, \perp\}$ - the current status of the node, initialised by $\perp$.

Finally, the challenger manages the following set of registers, which indicate $\mathcal{A}$'s compromise of secrets:

- long-term symmetric keys $\{\mathsf{LSKflag}_1, \ldots, \mathsf{LSKflag}_{n_N}\}$, where $\mathsf{LSKflag}_i \in \{\texttt{corrupt}, \texttt{clean}, \perp\}\ \forall\ i \in [n_N]$;
- per-stage secret state $\{\mathsf{PSSflag}_1^1, \mathsf{PSSflag}_2^1, \ldots, \mathsf{PSSflag}_{n_S}^1, \ldots \mathsf{PSSflag}_1^{n_N}, \mathsf{PSSflag}_2^{n_N}, \ldots, \mathsf{PSSflag}_{n_S}^{n_N}\}$ where $\forall i \in n_N,\ j \in n_S,\ \mathsf{PSSflag}_j^i \in \{\texttt{corrupt}, \texttt{clean}, \perp\}$;
- session keys $\{\mathsf{SKflag}_1^1, \mathsf{SKflag}_2^1, \ldots, \mathsf{SKflag}_{n_S}^1, \ldots \mathsf{SKflag}_1^{n_N}, \mathsf{SKflag}_2^{n_N}, \ldots, \mathsf{SKflag}_{n_S}^{n_N}\}$ where $\forall i \in n_N,\ j \in n_S,\ \mathsf{SKflag}_j^i \in \{\texttt{corrupt}, \texttt{clean}, \perp\}$.

7.5.2 Adversarial Interaction

In the game, the adversary $\mathcal{A}$ is able to communicate with the challenger and thus interact with the parties/sessions via the following set of queries:

- **Register**$(\lambda) \rightarrow id$: Allows $\mathcal{A}$ to register a new node with security parameters λ and gives $\mathcal{A}$ an identifier for the node id (which we denote N_{id}). For protocols where nodes do not have a public identifier, the index of the node is given to $\mathcal{A}$.
- **NextKey**$(\lambda, id) \rightarrow m$: Allows $\mathcal{A}$ to indicate that the node with public identifier id should attempt a new key agreement using (potentially) the new/updated security parameters λ. The challenger then returns any protocol messages m.
- **Corrupt**$(id) \rightarrow ltk$: Allows $\mathcal{A}$ to compromise the long-term key of the node $\pi_{id}.ltk$ with public identifier id.
- **Reveal**$(id, stid) \rightarrow sk$: Allows $\mathcal{A}$ to compromise the session key established between the hub node and the node N_{id} in stage $stid$. Note that $stid$ indicates the index of the session key established between the node id and the hub node. The challenger responds with the session key $\pi_i^s.sk$.
- **StateReveal**$(id, stid) \rightarrow psstate$: Allows $\mathcal{A}$ to compromise the per-stage secret state $psstate$ of the node with public identifier id. Note that $stid$ indicates the index of the stage-specific state, and the challenger responds with $\pi_i^s.psstate$.
- **Send**$(id, m) \rightarrow m'$: Allows $\mathcal{A}$ to send a message m to the node with identifier id currently running a protocol execution. Note that the node will update its per-session variables and potentially output a new message m'.

- **Test**$(id, stid) \rightarrow sk$: If the node N_{id} has completed its *stid*-stage key agreement, then the challenger uses the randomly-sampled bit $b \in \{0, 1\}$. If $b = 0$ the challenger responds with $\pi_i^s.sk$, otherwise the challenger responds with a random key from the same distribution.

We now formalise the advantage of a PPT algorithm $\mathcal{A}$ in winning the PPKA key-indistinguishability game.

Definition 7 (Key Indistinguishability)**.** *Let* Π *be a PPKA protocol and* n_N, $n_S \in \mathbb{N}$*. For a given cleanness predicate* clean*, and a PPT algorithm* $\mathcal{A}$*, we define the advantage of* $\mathcal{A}$ *in the key-indistinguishability game to be:*

$$\mathsf{Adv}^{\mathsf{PPKA\text{-}IND,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda) = |2 \cdot (\Pr[\mathsf{Exp}^{\mathsf{PPKA\text{-}IND,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda) = 1] - \frac{1}{2}|).$$

We say that Π *is* PPKA-IND*-secure if, for all* $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{PPKA\text{-}IND,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible in security parameter* λ*.*

7.5.3 Unlinkability

The experiments for PPKA key-indistinguishability and unlinkability are mostly identical. However, instead of using the **Test**$(id, stid)$ query, at some point $\mathcal{A}$ will stop and output (id_0, id_1). When $\mathcal{A}$ outputs (id_0, id_1), $\mathcal{C}$ runs **NextKey**(λ, id_0) and responds to queries as before. We will refer to this as the "challenge" node. However, when $\pi^{stid}_{id_0}.\alpha \leftarrow \texttt{accept}$, $\mathcal{C}$ then refers to the random bit b sampled at the beginning of the experiment and:

- if $b = 0$, then $\mathcal{C}$ runs **NextKey**(λ, id_0);
- if $b = 1$, then $\mathcal{C}$ runs **NextKey**(λ, id_1) instead.

$\mathcal{A}$ now uses the **SendTest**(m) query to send messages to the node N_{id_b} in order to avoid trivial identification. We will refer to this as the "unnamed node". $\mathcal{A}$ at some point terminates and outputs a guess bit b'. If $b' = 0$, then $\mathcal{A}$ is indicating that the unnamed node N_{id_b} was linked to the challenge node N_{id_0}. If $b' = 1$, then $\mathcal{A}$ is indicating that the unnamed node N_{id_b} was not linked to the challenge node N_{id_0}.

We now formalise the advantage of a PPT algorithm $\mathcal{A}$ in winning the PPKA unlinkability game.

Definition 8 (Unlinkability). *Let* Π *be a PPKA protocol, and* n_N, $n_S \in \mathbb{N}$. *For a given cleanness predicate* clean, *and a PPT algorithm* $\mathcal{A}$, *we define the advantage of* $\mathcal{A}$ *in the unlinkability game to be:*

$$\mathsf{Adv}^{\mathsf{PPKA\text{-}U,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda) = |2 \cdot (\Pr[\mathsf{Exp}^{\mathsf{PPKA\text{-}U,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda) = 1] - \frac{1}{2})|.$$

We say that Π *is* PPKA-U*-secure if, for all* $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{PPKA\text{-}U,clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible in the security parameter* λ.

7.5.4 Cleanness Predicates

The cleanness predicates are used in the security experiments to define the exact combination of secrets that $\mathcal{A}$ is able to compromise without trivially breaking the PPKA protocol. In order to capture KCI attacks and PrFS notions, we allow $\mathcal{A}$ to leak the long-term secret key of the "normal" nodes if $\mathcal{A}$ has not also leaked any previously established per-stage secret state. Our analysis is focused primarily on the normal nodes. We do not allow the compromise of the HN's long-term secrets, as security in all stages is lost in this scenario. We additionally describe a cleanness predicate for PPKA protocols that do not achieve PrFS or KCI resilience.

Definition 9 (PrFS-KCI-clean). *A session* π_i^s *such that* $\pi_i^s.\alpha =$ `accept` *in the* PPKA-IND *experiment defined in Figure 7.5 is* PrFS-KCI-clean *if* $\mathsf{SKflag}_{id}^{stid} \neq$ `corrupt` *and if* $\mathsf{LSKflag}_{id} =$ `corrupt` *then* $\forall\ s \leq stid$ $\mathsf{PSSflag}_{id}^{s} \neq$ `corrupt`.

Definition 10 (nPrFS-clean). *A session* π_i^s *such that* $\pi_i^s.\alpha =$ `accept` *in the* PPKA-IND *experiment defined in Figure 7.5 is* nPrFS-clean *if* $\mathsf{SKflag}_{id}^{stid} \neq$ `corrupt`.

Finally, we describe a cleanness predicate for our unlinkability game. It is straightforward to realise that if **Corrupt**(id_0) or **Corrupt**(id_1) were to be issued, it would trivially allow $\mathcal{A}$ to win in either of our PPKA protocols by simply reconstructing the tid_N field sent by the unnamed node. Similarly, we cannot allow the adversary to reveal the per-stage secret state for the current stage *stid* of the unnamed node N_{id_b}.

Definition 11 (U-clean). *A session* π_i^s *in the* PPKA-U *experiment defined in Figure 7.5 is* U-clean *if* $\mathsf{LSKflag}_{id} \neq$ `corrupt` *and* $\mathsf{PSSflag}_{id}^{stid} \neq$ `corrupt`.

$\mathsf{Exp}^{\mathsf{PPKA\text{-}IND},\mathsf{clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$:

$b \xleftarrow{\$} \{0,1\}$
tested $\leftarrow$ `false`
$k_{HN} \xleftarrow{\$} \mathsf{HKeyGen}(\lambda)$
$\mathsf{LSKflag}_i, \ldots, \mathsf{LSKflag}_{n_N} \leftarrow$ `clean`
$\mathsf{PSSflag}^1_1, \ldots, \mathsf{PSSflag}^{n_N}_{n_S} \leftarrow$ `clean`
$\mathsf{SKflag}^1_1, \ldots, \mathsf{SKflag}^{n_N}_{n_S} \leftarrow$ `clean`
$ctr \leftarrow 0$
$b' \xleftarrow{\$} \mathcal{A}^{\mathsf{Send},\mathsf{Register},\mathsf{NextKey},\mathsf{Corrupt},\star\mathsf{Reveal},\mathsf{Test}}(\lambda)$
For $(id, stid)$ **such that Test**$(id, stid)$ **was issued**:
if $\mathsf{clean}(\pi^s_i)$ **then**
  return $(b = b')$
else
  return $b' \xleftarrow{\$} \{0,1\}$
end if

Reveal$(id, stid)$:

if $\pi^s_i.\alpha \neq$ `accept` **then**
  return $\perp$
end if
$\mathsf{SKflag}^{id}_{stid} \leftarrow$ `corrupt`
return $\pi^s_i.sk$

StateReveal$(id, stid)$:

if $\pi^s_i.psstate = \perp$ **then**
  return $\perp$
end if
$\mathsf{PSSflag}^{id}_{stid} \leftarrow$ `corrupt`
return $\pi^s_i.psstate$

NextKey(λ, id):

let $stid = max\{s : \pi^s_{id}.\alpha \neq \perp\}$
if $(\pi^s_i.\alpha \neq$ `accept`$)$ **then**
  return $\perp$
end if
$stid \leftarrow stid + 1$
$\pi^s_i.\alpha \leftarrow$ `active`
$\pi^s_i, m' \leftarrow \Pi.\mathsf{NF}(\lambda, \pi^s_i, \perp)$
return m'

Send(id, m):

if $id = HN$ **then**
  return $\Pi.\mathsf{HF}(\lambda, k_{HN}, m)$
end if
let $stid = max\{s : \pi^s_{id}.\alpha \neq \perp\}$
if $\pi^s_i.\alpha \neq$ `active` **then**
  return $\perp$
end if
$\pi^s_i.m_r \leftarrow \pi^s_i.m_r \| m$
$(\pi^s_i, m') \leftarrow \Pi.\mathsf{NF}(\lambda, \pi^s_i, m)$
$\pi^s_i.m_s \leftarrow \pi^s_i.m_s \| m'$
if $\pi^s_i.\alpha \leftarrow$ `accept` **then**
  $\pi^{stid+1}_{id}.psstate \leftarrow \mathsf{StateUpdate}(\lambda, \pi^s_i)$
end if
return m'

Corrupt(id):

$\mathsf{LSKflag}_{id} \leftarrow$ `corrupt`
return $\pi_{id}.ltk$

Test$(id, stid)$:

if (tested $=$ `true`$) \vee (\pi^s_i.\alpha \neq$ `accept`$)$ **then**
  return $\perp$
end if
tested $\leftarrow$ `true`
if $b = 0$ **then**
  return $\pi^s_i.sk$
else
  $sk \xleftarrow{\$} \mathcal{K}$
  return sk
end if

Register(λ):

$ctr \leftarrow ctr + 1$
$\pi.stid \leftarrow 1$
$\pi.ltk \leftarrow \Pi.\mathsf{NKeyGen}(\lambda)$
$\pi.id \leftarrow ctr$
$\pi.psstate \leftarrow \Pi.\mathsf{StateGen}(\lambda, k_{HN}, \pi.ltk)$
return $\pi.id$

$\mathsf{Exp}^{\mathsf{PPKA\text{-}U},\mathsf{clean}}_{\Pi,n_N,n_S,\mathcal{A}}(\lambda)$:

$b \xleftarrow{\$} \{0,1\}$
$k_{HN} \xleftarrow{\$} \mathsf{HKeyGen}(\lambda)$
$\mathsf{LSKflag}_i, \ldots, \mathsf{LSKflag}_{n_N} \leftarrow$ `clean`
$\mathsf{PSSflag}^1_1, \ldots, \mathsf{PSSflag}^{n_N}_{n_S} \leftarrow$ `clean`
$\mathsf{SKflag}^1_1, \ldots, \mathsf{SKflag}^{n_N}_{n_S} \leftarrow$ `clean`
$ctr \leftarrow 0$
$(id_0, id_1) \xleftarrow{\$} \mathcal{A}^{\mathsf{Send},\mathsf{Register},\mathsf{NextKey},\mathsf{Corrupt},\star\mathsf{Reveal}}(\lambda)$
NextKey$(\lambda, id_0) \rightarrow m$
$\emptyset \leftarrow \mathcal{A}^{\mathsf{Send},\mathsf{Register},\mathsf{NextKey},\mathsf{Corrupt},\star\mathsf{Reveal}}(\lambda, m)$
if $\pi^{stid}_{id_0}.\alpha \leftarrow$ `accept` **then**
  NextKey$(\lambda, id_b) \rightarrow m'$
end if
$b' \xleftarrow{\$} \mathcal{A}^{\mathsf{Send},\mathsf{Register},\mathsf{NextKey},\mathsf{Corrupt},\star\mathsf{Reveal},\mathsf{SendTest}}(\lambda, m')$
if $\mathsf{clean}(\pi^{stid_b}_{id_0}) \wedge \mathsf{clean}(\pi^{stid_b}_{id_1})$ **then**
  return $(b = b')$
else
  return $b' \xleftarrow{\$} \{0,1\}$
end if

SendTest(m):

$\mathsf{Send}(id_b, m) \rightarrow m'$
return m'

Figure 7.5: An algorithmic description of the PPKA-IND and PPKA-U security experiments.

7.6 Analysis of the PPKA Protocols

7.6.1 Security and Privacy Analysis

Before we begin, we show that an adversary $\mathcal{A}$ is unable to recover HN's long-term secret k_{HN} (with non-negligible probability) even if $\mathcal{A}$ reveals all long-term secrets id_N of all nodes and all per-stage secret states $psstate$. In our proofs we work within the random oracle model, and $\mathcal{A}$ cannot learn anything about k_{HN} from hash outputs $h(k_{HN}, X)$ (where X is any concatenation of arbitrary values). We turn to $\mathcal{A}$ attempting to learn k_{HN} that has been "blinded" through exclusive-or (XOR) operations. The generic construction of messages that include k_{HN} is as follows:

- $b_N = k_{HN} \oplus k_N \oplus id_N \oplus h(k_{HN}, k_N)$;
- $\mu = k_{HN} \oplus k_N^+ \oplus id_N \oplus h(k_{HN}, k_N^+) \oplus h(id_N, t_N, r_N, id'_N) \oplus f_N \oplus r_N$.

Taking μ first, we note that k_N^+ (independently sampled by HN, uniformly-at-random, in each stage) acts as the key in a one-time-pad, perfectly hiding the long-term secret key k_{HN} of HN, the long-term secret key id_N of N_{id} and the value $h(k_{HN}, k_N^+)$. As k_N^+ is an internal value that is known only to the challenger implementing HN, it cannot be compromised by $\mathcal{A}$ via **Reveal**, **Corrupt** or **StateReveal** queries. For b_N, we note that k_N (randomly sampled by HN in a previous stage) is still acting as the same key k_N^+ in a one-time-pad, and thus still perfectly hiding the same message; i.e. the long-term secret key k_{HN} of HN, the long-term secret key id_N of N_{id} and the value $h(k_{HN}, k_N)$. We argue then that $\mathcal{A}$ cannot recover k_{HN}. We can further conclude that an adversary that compromises fewer internal states and long-term secret keys will also be unable to recompute k_{HN}. We can continue our proof knowing that the best strategy for $\mathcal{A}$ to recover k_{HN} is to attempt to brute-force the value.

We now show that an adversary $\mathcal{A}$ that does not issue a **Corrupt**(id) query cannot recover the long-term secret key id_N of node N_{id}. As before, we note that, since we instantiate the hash function as a random oracle, the adversary cannot invert hash outputs of the form $h(id_N, X)$ (where X is some arbitrary concatenation of values) in order to learn id_N. We can now focus on the adversary attempting to learn id_N from "blinded" values by XORing them with other values. In each stage of the protocol execution, this is available to $\mathcal{A}$ in the following generic ways:

- $a_N = id_N \oplus h(k_{HN}, k_N)$;
- $b_N = k_{HN} \oplus k_N \oplus id_N \oplus h(k_{HN}, k_N)$;
- $\eta = r_N \oplus f_N \oplus id_N \oplus k_{HN} \oplus k_N \oplus h(id_N, f_N) \oplus h(k_{HN}, k_N^+)$;
- $\mu = r_N \oplus f_N \oplus id_N \oplus k_{HN} \oplus k_N^+ \oplus h(id_N, t_N, r_N, id'_N) \oplus h(k_{HN}, k_N^+)$.

If this is the first stage of the protocol execution for node N_{id}, then a_N and b_N are established in some out-of-band way. Thus $h(k_{HN}, k_N)$ and k_N act as uniformly random and independent keys in a one-time pad, perfectly hiding id_N and $k_{HN} \oplus id_N \oplus h(k_{HN}, k_N)$ (for a_N and b_N respectively). Since, by the previous argument, the best strategy for $\mathcal{A}$ to recover k_{HN} is simply to guess (and we instantiate the hash function with a random oracle), in order to recompute $h(k_{HN}, k_N)$ $\mathcal{A}$ must either guess k_{HN} or to guess $h(k_{HN}, k_N)$. Since they are the same bit-length, the probability of $\mathcal{A}$ doing either is the same: $2^{-\lambda}$.

If this is not the first stage of the protocol execution, then a_N and b_N were sent as "sub-XOR" of a previous stage η and μ. We argue that $h(k_{HN}, k_N^+)$ and k_N^+ act as keys to one-time-pads for η and μ respectively, and remain the keys to the one-time-pad perfectly hiding id_N and $k_{HN} \oplus id_N \oplus h(k_{HN}, k_N)$ (for a_N and b_N respectively) in the following stage. It follows then that the best strategy $\mathcal{A}$ has in recovering id_N is to merely guess $id_N : 2^{-\lambda}$.

We now prove the key-indistinguishability of our PPKA protocols given in Figures 7.3 and 7.4. We begin with PPKA-2, as it captures the strongest notions of security, capturing PrFS, KCI resilience, key randomness, known key security and authentication. Afterwards, we turn to proving the unlinkability of PPKA-2. We then prove key indistinguishability and unlinkability of PPKA-1. As PPKA-1 is essentially a truncated version of PPKA-2, this allows us to omit the most repetitive details of the proofs.

Theorem 2 (Key Indistinguishability of PPKA-2)**.** *The privacy-preserving key agreement protocol* PPKA-2 *given in Figure 7.4 is* PPKA-IND*-secure with cleanness predicate* PrFS-KCI-clean *(capturing PrFS and KCI resilience) and assuming all hash functions are random oracles. For any PPT algorithm $\mathcal{A}$ against the* PPKA-IND *key indistinguishability game,* $\mathsf{Adv}^{\text{PPKA-IND,PrFS-KCI-clean}}_{\text{PPKA-2},n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible in the security parameter λ.*

Proof. For our proof, we assume that a test query **Test**$(id, stid)$ has been issued, and separate into the following three cases:

1. π_i^s has accepted such that $\pi_i^s.m_r \neq \mathsf{PPKA\text{-}2.HF}(\lambda, k_{HN}, \pi_i^s.m_s)$;
2. π_i^s has accepted such that $\pi_i^s.m_r = \mathsf{PPKA\text{-}2.HF}(\lambda, k_{HN}, \pi_i^s.m_s)$ and $\mathbf{Corrupt}(id)$ has not been issued;
3. π_i^s has accepted such that $\pi_i^s.m_r = \mathsf{PPKA\text{-}2.HF}(\lambda, k_{HN}, \pi_i^s.m_s)$ and $\mathbf{Corrupt}(id)$ has been issued; by the definition of the cleanness predicate $\mathsf{PrFS\text{-}KCI\text{-}clean}$, we assume that the per-stage secret state has not been revealed for any stage $s \leq stid$.

Case 1. In this case we show that once a $\mathbf{Test}(id, stid)$ query is issued, the probability, the session π_i^s sets $\pi_i^s.\alpha \leftarrow \texttt{accept}$ such that $\pi_i^s.m_r \neq \mathsf{PPKA\text{-}2.HF}(\lambda, k_{HN}, \pi_i^s.m_s)$, is negligible.

Game 0

This is a normal PPKA key-indistinguishability game. Thus we have:

$$\mathsf{Adv}^{\mathsf{PPKA\text{-}IND},C_1}_{\mathsf{PPKA\text{-}2},n_N,n_S,\mathcal{A}}(\lambda) = \Pr(break_0).$$

Game 1

In this game, we guess the index $(id, stid)$ of the session π_i^s, and abort if, during the execution of the experiment, a query $\mathbf{Test}(i^*, s^*)$ is received and $(i^*, s^*) \neq (id, stid)$. Thus we have:

$$\Pr(break_0) \leq n_N n_S \cdot \Pr(break_1).$$

Game 2

In this game, we replace the $h(k_{HN}, k_N)$ value computed within π_i^s (and, potentially, in the hub node processing $\pi_i^s.m_s$) with a uniformly-random value $\widetilde{h(k_{HN}, k_N)}$. We note that since we instantiate the hash function with a random oracle, the distribution is identical to $h(k_{HN}, k_N)$. Thus the only way that $\mathcal{A}$ can detect this change is to query (k_{HN}, k_N) to the random oracle. Since the only way for $\mathcal{A}$ to do this is to recover k_{HN} fully, and we argued previously that $\mathcal{A}$'s probability of success in this endeavour is $2^{-\lambda}$, we have:

$$\Pr(break_1) \leq 2^{-\lambda} + \Pr(break_2).$$

Game 3

In this game we argue that the adversary $\mathcal{A}$ has a negligible probability of producing a value $\hat{\beta} = h(h(\widetilde{k_{HN}, k_N}),\ \hat{z_N}, \hat{r_N}, \hat{f_N}, \hat{\delta}, \hat{\eta}, \hat{\mu}, \hat{id'_N})$. Note that for $\pi_i^s.\alpha$ to reach `accept`, $\mathcal{A}$ must produce such a value $\hat{\beta}$. We know by the definition of Case 1 that the following must be true:

$$\pi_i^s.m_r = \langle \hat{\alpha}, \hat{\beta}, \hat{\eta}, \hat{\mu}, \hat{\delta}, \hat{id'_N} \rangle \neq \mathsf{PPKA\text{-}2.HF}(\lambda, k_{HN}, \pi_i^s.m_s).$$

Since all message fields are included in the computation of $\hat{\beta}$, and the message received by the test session does not match any output from an honest hub node, we know that the only way that $\mathcal{A}$ can cause π_i^s to reach `accept` is to query $h(\widetilde{k_{HN}, k_N}), \hat{z_N}, \hat{r_N}, \hat{f_N}, \hat{\delta}, \hat{\eta}, \hat{\mu}, \hat{id'_N}$ to the random oracle. However, since, by Game 2, $h(\widetilde{k_{HN}, k_N})$ is a uniformly-random value sampled independently from the protocol flow, the only way for $\mathcal{A}$ to produce such an input is to guess $h(\widetilde{k_{HN}, k_N})$. Thus we have:

$$\Pr(break_2) \leq 2^{-\lambda} + \Pr(break_3).$$

It is clear that if the session π_i^s such that $\mathbf{Test}(id, stid)$ must be issued (by Game 1) cannot reach $\pi_i^s.\alpha \leftarrow$ `accept`, then in Game 3 the experiment proceeds identically regardless of the bit b sampled by the challenger. Thus:

$$\Pr(break_3) = 0.$$

Case 2. In this case, we show that an adversary who issues a $\mathbf{Test}(stid, id)$ query (and does not also issue a $\mathbf{Corrupt}(id)$ query) cannot win the key-indistinguishability game with non-negligible probability.

Game 0

This is a normal PPKA key-indistinguishability game. Thus we have:

$$\mathsf{Adv}^{\mathsf{PPKA\text{-}IND},C_2}_{\mathsf{PPKA\text{-}2},n_N,n_S,\mathcal{A}}(\lambda) = \Pr(break_0).$$

Game 1

In this game, we guess the index $(id, stid)$ of the session π_i^s and abort if, during the execution of the experiment, a query $\mathbf{Test}(i^*, s^*)$ is received and $(i^*, s^*) \neq (id, stid)$. Thus we have:

$$\Pr(break_0) \leq n_N n_S \cdot \Pr(break_1).$$

Game 2

In this game, we replace the session key k_S computed by the node N_{id} in stage $stid$ with a uniformly-random and independent value $\widetilde{k_S}$. First we note that k_S is computed as $k_S = h(id_N, z_N, r_N, f_N, x_N)$. Since we instantiate the hash function as a random oracle, the distributions of $\widetilde{k_S}$ and k_S are identical. In order to distinguish this change, $\mathcal{A}$ must be able to query the random oracle with the input (id_N, r_N, f_N, x_N). Since we argued previously that in order to recover id_N (the long-term secret key of the node N_{id}), $\mathcal{A}$'s only strategy to distinguish this change would be to guess the long-term secret id_N. The probability of $\mathcal{A}$ distinguishing this replacement is $2^{-\lambda}$, where λ is the bit-length of id_N.

After this change, the session key returned to $\mathcal{A}$ as the response to the **Test**$(stid, id)$ query is a uniformly-random value independent of the protocol execution, regardless of the bit b sampled by the challenger. Thus we have:

$$\Pr(break_1) \leq 2^{-\lambda}.$$

Case 3. In this case, we show that an adversary who issues a **Test**$(stid, id)$ query (and does not issue **StateReveal** queries for all per-stage secret states established before stage $stid$) cannot win the key-indistinguishability game.

Game 0

This is a normal PPKA key-indistinguishability game. Thus we have:

$$\mathsf{Adv}^{\mathsf{PPKA\text{-}IND},C_3}_{\mathsf{PPKA}\text{-}2,n_N,n_S,\mathcal{A}}(\lambda) = \Pr(break_0).$$

Game 1

In this game, we guess the index $(id, stid)$ of the session π_i^s and abort if, during the execution of the experiment, a query **Test**(i^*, s^*) is received and $(i^*, s^*) \neq (id, stid)$. Thus we have:

$$\Pr(break_0) \leq n_N n_S \cdot \Pr(break_1).$$

Game 2

In this game, we replace the $z_N = h(k_{HN}, id_N, k_N)$ value held in secret stage by the node N_{id} with a uniformly random value $\widetilde{z_N}$ independent from the protocol execution. Since we instantiate the hash function with a random oracle, the distributions of z_N and $\widetilde{z_N}$ are identical. Thus, in order to detect this change, $\mathcal{A}$ must query the random oracle with the input k_{HN}, id_N, k_N. Since, by earlier arguments, the best strategy $\mathcal{A}$ has to recover k_{HN} is simply to guess k_{HN}, the probability that $\mathcal{A}$ is able to do this is $2^{-\lambda}$. Thus:

$$\Pr(break_1) = 2^{-\lambda} + \Pr(break_2).$$

Game 3

In this game, we replace the computation of the z_N^+ encryption key $k_z = h(\widetilde{z_N}, id_N, r_N, f_N, 0)$ with a uniformly-random and independent value $\widetilde{k_z}$. We note that since $\widetilde{z_N}$ (by Game 2) is already a uniformly random value, and the hash function is instantiated with a random oracle, this replacement is sound and indistinguishable from the perspective of $\mathcal{A}$. Thus:

$$\Pr(break_2) = \Pr(break_3).$$

Game 4

In this game, we replace the contents of ciphertext δ with a random string of the same length, and abort if the ciphertext δ sent by the hub node *HN* is not the ciphertext received by N_{id}, and the output of decrypting δ is not $\perp$. We do so by constructing an algorithm $\mathcal{B}$ that interacts with an IND-CCA challenger in the following way: $\mathcal{B}$ acts identically as in Game 3, except for the hub node protocol execution that computes $\widetilde{k_z}$. Instead, when $\mathcal{B}$ computes δ, $\mathcal{B}$ selects a uniformly-random string $\widetilde{z_N^+}$ (of the same length as z_N^+) and submits $(z_N^+, \widetilde{z_N^+})$ to the IND-CCA encryption oracle Enc.

When the random bit b sampled by the IND-CCA challenger is 0, then δ contains the encryption of z_N^+, so $\mathcal{B}$ is a perfect simulation of Game 3. However, when the bit b sampled by the IND-CCA challenger is 1, then δ contains a random string $\widetilde{z_N^+}$ and thus $\mathcal{B}$ is a perfect simulator of Game 4. Since, in Game 3, the z_N^+ encryption

key $\widetilde{k_z}$ is uniformly-random and independent of the protocol execution, this replacement is sound. Any adversary capable of distinguishing this change can break the confidentiality of the IND-CCA encryption scheme and guess b with perfect success. Thus:

$$\Pr(break_3) \leq \mathsf{Adv}_{\mathsf{Enc}}^{\mathsf{IND\text{-}CCA}} + \Pr(break_4).$$

Game 5

We now note that by Game 4, z_N^+ has been established in an out-of-band way, reminiscent of the first stage run by node N_{id}. We now repeat the process of Games 2, 3, and 4 $(stid - 2)$ times to establish a z_N value for stage $stid$ run by node N_{id} that is indistinguishable from establishing z_N in some out-of-band way. Thus:

$$\Pr(break_4) \leq (stid - 2) \cdot (2^{-\lambda} + \mathsf{Adv}_{\mathsf{Enc}}^{\mathsf{IND\text{-}CCA}}) + \Pr(break_5).$$

Game 6

We replace z_N with a uniformly-random and independent value $\widetilde{z_N}$ in stage $stid$ of node N_{id} by the same argument as Game 2. Thus:

$$\Pr(break_5) = 2^{-\lambda} + \Pr(break_6).$$

Game 7

In this game, we replace the computation of the session key $k_s = h(id_N, \widetilde{z_N}, r_N, f_N, 1)$ with a uniformly-random and independent value $\widetilde{k_s}$. We note that since $\widetilde{z_N}$ (by Game 6) is already a uniformly random value, and the hash function is instantiated with a random oracle, this replacement is sound and indistinguishable from the perspective of $\mathcal{A}$. Thus $\Pr(break_6) = \Pr(break_7)$. We finally note that the session key established by π_i^s is now uniformly random and independent of the protocol flow, and of the bit b sampled by the PPKA-IND challenger. Thus:

$$\Pr(break_7) = 0.$$

We follow our proof of the key-indistinguishability of PPKA-2 by proving the unlinkability of PPKA-2.

Theorem 3 (Unlinkability of PPKA-2)**.** *The* PPKA-2 *given in Figure 7.4 is* PPKA-U*-secure with cleanness predicate* U-clean *and assuming all hash functions are random oracles. For any PPT algorithm $\mathcal{A}$ against the* PPKA-U *unlinkability game described in Figure 7.5,* $\mathsf{Adv}^{\text{PPKA-U,U-clean}}_{\text{PPKA-2},n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible in the security parameter λ.*

Proof. We begin by restating the U-clean cleanness predicate, and reiterating the impact upon our proof. For both nodes N_{id_0} and N_{id_1}, we know that the queries **Corrupt**(id_0) and **Corrupt**(id_1) have not been issued. In addition, for the stage $stid_b$ run by the unnamed node N_{id_b}, we know that a **StateReveal**$(id_b)(stid_b)$ query has not been issued.

Game 0

This is a normal PPKA unlinkability game. Thus we have:

$$\mathsf{Adv}^{\text{PPKA-U}}_{\text{PPKA-2},n_N,n_S,\mathcal{A}}(\lambda) = \Pr(break_0).$$

Game 1

In this game, in the unnamed session $\pi^{stid_b}_{id_b}$, we replace the hash outputs of the form $h(id_N, X)$ (where X is a concatenation of arbitrary stings) with a uniformly random values $\widetilde{h(id_N, X)}$ chosen independently of the protocol flow. As before, since we instantiate (in our proof) the hash function with a random oracle, the distribution of this change is indistinguishable. In order to detect this change, $\mathcal{A}$ must query the random oracle with the input (id_N, X). As per our previous arguments, in order to query id_N to the random oracle, $\mathcal{A}$ must first recover id_N. Since the best strategy to recover id_N is simply to guess the value of id_N, the probability of $\mathcal{A}$ distinguishing this change is $2^{-\lambda}$. Thus we have:

$$\Pr(break_0) = 2^{-\lambda} + \Pr(break_1).$$

Game 2

In this game, in the unnamed session $\pi^{stid_b}_{id_b}$, we replace the hash outputs of the form $h(k_{HN}, X)$ (where X is either k_N or k^+_N) with uniformly random values $\widetilde{h(k_{HN}, X)}$ chosen independently of the protocol flow. As before, since we instantiate (in our proof) the hash function with a random oracle, the distributions of Game 1 and Game 2 are indistinguishable. In order to detect this change, $\mathcal{A}$ must query the

random oracle with the input (k_{HN}, X). As per our previous arguments, in order to query k_{HN} to the random oracle, $\mathcal{A}$ must first recover k_{HN}. Since the best strategy to recover k_{HN} is simply to guess the value of k_{HN}, the probability of $\mathcal{A}$ distinguishing this change is $2^{-\lambda}$. Thus we have:

$$\Pr(break_1) = 2^{-\lambda} + \Pr(break_2).$$

Game 3

In this game, in the message output by the hub node for the unnamed session $\pi_{id_b}^{stid_b}$, we replace the hash outputs $\beta = h(\widetilde{h(k_{HN}, k_N^+)}, z_N, r_N, f_N, \delta, \eta, \mu, id'_N)$ with a uniformly random value $\widetilde{\beta}$ chosen independently of the protocol flow. As previous arguments, the distributions of Game 2 and Game 3 are indistinguishable. In order to detect this change, $\mathcal{A}$ must query the random oracle with the input $(\widetilde{h(k_{HN}, k_N^+)}, z_N, r_N, f_N, \delta, \eta, \mu, id'_N)$. Since $\widetilde{h(k_{HN}, k_N^+)}$ is already a uniformly random value independent of the protocol flow (by Game 2), the best strategy to distinguish this change is to simply guess the value of $\widetilde{h(k_{HN}, k_N^+)}$. Thus we have:

$$\Pr(break_2) = 2^{-\lambda} + \Pr(break_3).$$

Game 4

In this game, in the unnamed session $\pi_{id_b}^{stid_b}$ we replace the computation of the z_N^+ key $k_z = h(z_N, id_N, r_N, f_N, 0)$ with a uniformly-random and independent value $\widetilde{k_z}$. We note that, since we instantiate the hash function with a random oracle, the distributions of $\widetilde{k_z}$ and k_z are indistinguishable. Thus, in order to detect this change, $\mathcal{A}$ must query the random oracle with the input $z_N, id_N, r_N, f_N, 0$. By earlier arguments, the best strategy $\mathcal{A}$ has to recover id_N is simply to guess id_N. Thus:

$$\Pr(break_3) = 2^{-\lambda} + \Pr(break_4).$$

Game 5

In this game we replace the value δ sent by the hub node to the unnamed session $\pi_{id_b}^{stid_b}$ with a uniformly-random and independent value $\widetilde{\delta} \xleftarrow{\$} \{0,1\}^\lambda$. We do so by constructing an algorithm $\mathcal{B}$ that interacts with a PRF challenger in the following way, $\mathcal{B}$ acts identically as in Game 4, expect for the hub node protocol execution that computes $\widetilde{k_z}$. Instead, $\mathcal{B}$ initialise a PRF challenger and queries (z_n^+), and uses

the output $\widetilde{\delta}$ from the PRF challenger to replace the computation of δ. Since, by Game 4, $\widetilde{k_z}$ is a uniformly random and independent value, this replacement is sound. If the test bit sampled by the PRF challenger is 0, then $\widetilde{\delta} \leftarrow \mathsf{Enc}(\widetilde{k_z}, z_N^+)$ and we are in Game 4. If the test bit sampled by the PRF challenger is 1, then $\widetilde{\delta} \xleftarrow{\$} \{0,1\}^\lambda$ and we are in Game 5. Thus any adversary $\mathcal{A}$ capable of distinguishing this change can be turned into a successful adversary against the PRF security of the encryption scheme Enc, and we find:

$$\Pr(break_4) \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{Enc},\mathcal{A}}(\lambda) + \Pr(break_5).$$

We pause here to reflect on the consequences of these changes. The first message sent by the unnamed node is $\langle \widetilde{tid_N}, y_N, a_N, b_N, t_N, id'_N \rangle$. Since t_N is a timestamp and id'_N is sampled identically regardless of the identity of the unnamed node, the distributions of these fields are similarly identical, independent of the choice of the randomly sampled bit b. $\widetilde{tid_N}$ is a uniformly-random valued and independent of the protocol flow (by Game 1), as it is the output of a random oracle query that is of the form $(id_{N_b}, id'_N, t_n, r_n)$. This is true regardless of the choice of the randomly sampled bit b of the challenger.

For y_N we remark that r_N is a uniformly-random value sampled identically from the same distribution regardless of the node identity. This value acts as the key in a one-time-pad, perfectly hiding $h(\widetilde{k_{HN}, k_N})$. As r_N is not reused (as a key) in any message in any stage, thus y_N is a uniformly-random value, regardless of node identity. The value a_N is also a uniformly random. Here, $h(\widetilde{k_{HN}, k_N})$ acts as the key in a one-time-pad, perfectly hiding the long-term secret key id_N of the node by Game 2. Since $h(\widetilde{k_{HN}, k_N})$ is not reused (as a key) in any message in any stage, a_N is a uniformly random value, regardless of the node identity, or the bit b randomly sampled by the challenger. Finally, we turn to b_N. We note that this time k_N (randomly sampled by the hub node in a previous stage, uniformly-at-random) acts as the key in a one-time-pad, perfectly hiding the long-term secret key k_{HN} of the hub node, the long-term secret key id_N of the node and the value $h(\widetilde{k_{HN}, k_N})$. As k_N is not reused (as a key) in any message in any stage, thus b_N is a uniformly-random value, regardless of node identity.

We examine the first message received by the unnamed node, $\langle \alpha, \beta, \eta, \mu, \delta, id'_N \rangle$. Again, id'_N is sampled identically regardless of the identity of the unnamed node; the distributions of the fields are similarly identical independent of the choice of the randomly sampled bit b. For α we remark that f_N is a uniformly-random value

sampled identically from the same distribution regardless of the node identity. This value acts as the key in a one-time-pad, perfectly hiding $h(\widetilde{k_{HN}, k_N^+})$. As f_N is not reused (as a key) in any message in any stage, thus α is a uniformly-random value, regardless of node identity. Value η is also a uniformly-random. Here, $h(\widetilde{id_N, t_N})$ acts as the key in a one-time-pad, perfectly hiding the values r_N, f_N and a_N^+ by Game 1. Since $h(\widetilde{id_N, t_N})$ is not reused (as a key) in any message in any stage, η is a uniformly random value, independent of the node identity, or the bit b randomly sampled by the challenger.

A similar argument apples for μ, substituting $h(id_N, \widetilde{t_N, r_N}, id'_N)$ for $h(\widetilde{id_N, t_N})$. Value $\widetilde{\beta}$ is a uniformly-random and independent of the protocol flow (by Game 3), as it is the output of a random oracle query that is of the form $(h(\widetilde{k_{HN}, k_N^+}), z_N, r_N, f_N, \delta, \eta, \mu, id'_N)$. This is true regardless of the choice of the randomly sampled bit b of the challenger. Finally, we rely on the PRF security of the encryption scheme Enc to replace the δ field returned by the hub node. By Game 5, the value $\widetilde{\delta}$ is uniformly-random and independent of the protocol regardless of the node identity id_b. We note that all message fields have the same distribution regardless of the challenger's randomly-sampled bit b. Thus we have:

$$\Pr(break_5) = 0.$$

We now prove key-indistinguishability of our proposed PPKA-1, capturing known key security, and key randomness, but not forward-secrecy. It follows identically from *Case 2* of the proof of PPKA-2 key-indistinguishability, as it does not capture PrFS or KCI resilience. However, it still captures known key security, and key randomness and (obviously) key-indistinguishability.

Theorem 4 (Key Indistinguishability of PPKA-1)**.** *The* PPKA-1 *given in Figure 7.3 is* PPKA-IND*-secure with cleanness predicate* nPrFS-clean *(capturing neither PrFS nor KCI resilience) and assuming all hash functions are random oracles. For any PPT algorithm $\mathcal{A}$ against the* PPKA-IND *key-indistinguishability game,* $\mathsf{Adv}^{\mathsf{PPKA\text{-}IND},\mathsf{nPrFS\text{-}clean}}_{\mathsf{PPKA\text{-}1},n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible in the security parameter λ.*

Proof. For our proof, we note that we cannot prove PrFS or KCI resilience for the proposed PPKA-1. Thus, unlike PPKA-2, the cleanness predicate nPrFS-clean ensures that **Corrupt**(id) has not been issued. In this case, we assume that the per-stage secret state has been compromised at any (or perhaps, at all) previous stages. Since PPKA-1 sends the per-stage secret state $\langle a_N, b_N \rangle$ in the clear, this has no bearing on our security proof of PPKA-1.

Similarly to the proof for PPKA-2, we begin by showing that the adversary is unable to recover the HN secret key k_{HN} (with non-negligible probability) even if $\mathcal{A}$ completely reveals the long-term secret keys of every normal node and the per-stage secret states of the nodes. This argument follows identically to the argument for the secrecy of k_{HN} in the proof of PPKA-2, and we can continue our proof knowing that the best strategy $\mathcal{A}$ has for recovering k_{HN} is to guess k_{HN}.

In this proof, we show that an adversary which issues a **Test**$(stid, id)$ query (and does not also issue a **Corrupt**(id) query) cannot win the key-indistinguishability game with negligible probability. Before we begin, we show that an adversary who does not issue a **Corrupt**(id) query cannot recover the long-term secret key id_N of node N_{id}. This argument follows identically to the argument for the secrecy of id_N in the proof of PPKA-2, and we can continue our proof knowing that the best strategy $\mathcal{A}$ has for recovering id_N is to guess id_N.

Game 0

This is a normal PPKA key-indistinguishability game. Thus we have:

$$\mathsf{Adv}^{\mathsf{PPKA\text{-}IND},C_1}_{\mathsf{PPKA}\text{-}1,n_N,n_S,\mathcal{A}}(\lambda) = \Pr(break_0).$$

Game 1

In this game, we guess the index $(id, stid)$ of the session π_i^s, and abort if, during the execution of the experiment, a query **Test**(i^*, s^*) is received and $(i^*, s^*) \neq (id, stid)$. Thus we have:

$$\Pr(break_0) \leq n_N n_S \cdot \Pr(break_1).$$

Game 2

In this game, we replace the session key k_S computed by the node N_{id} in stage $stid$ with a uniformly-random and independent value $\widetilde{k_S}$. First we note that k_S is computed as $k_S = h(id_N, r_N, f_N, x_N)$. Since we instantiate the hash function as a random oracle, the distributions of $\widetilde{k_S}$ and k_S are identical thus, in order to distinguish this change, $\mathcal{A}$ must be able to query the random oracle with the input (id_N, r_N, f_N, x_N). We argued previously that in order to recover id_N (the long-term secret key of the node N_{id}), $\mathcal{A}$'s only strategy in distinguishing this change would be

to guess the long-term secret key id_N. Thus the probability of $\mathcal{A}$ in distinguishing this replacement is $2^{-\lambda}$, where λ is the bit-length of id_N.

After this change, the session key returned to $\mathcal{A}$ as the response to the **Test**$(stid, id)$ query is a uniformly-random value independent of the protocol execution, regardless of the bit b sampled by the challenger. Thus we have:

$$\Pr(break_1) \leq 2^{-\lambda} + 0.$$

Finally, we finish our security analysis by proving the unlinkability of PPKA-1.

Theorem 5 (Unlinkability of PPKA-1)**.** *The* PPKA-1 *given in Figure 7.3 is* PPKA-U*-secure with cleanness predicate* U-clean *and assuming all hash functions are random oracles. For any PPT algorithm $\mathcal{A}$ against the* PPKA-U *unlinkability game described in Figure 7.5,* $\mathsf{Adv}^{\text{PPKA-U,U-clean}}_{\text{PPKA-1},n_N,n_S,\mathcal{A}}(\lambda)$ *is negligible in the security parameter λ.*

Proof. The proof of the unlinkability of PPKA-1 is identical to the proof of unlinkability for PPKA-2, (with the exception of Game 4 and Game 5, since PPKA-1 does not have z_N state, nor a δ field in the hub node's response) and so we omit repeating it here.

7.6.2 Functional Analysis

The proposed PPKA protocols can easily be adapted for direct communication between N and HN by removal of Steps 2 and 4 from their respective *authentication phases*. As our PPKA protocols are also based on symmetric cryptographic primitives, they preserve the efficiency of the original scheme from a computation, communication and storage perspective without the aid of any additional network infrastructure. Moreover, in our protocols the timestamp field can be of any arbitrary length to suit the underlying protocol layers, unlike [103]. Assuming a B-bit hash digest and 16-bit pseudo-identity id'_N for node N, Tables 7.5 and 7.6 depict the various overheads associated with PPKA Protocols 1 and 2, respectively. In these tables, h denotes an instance of a hash operation and $\oplus$ denotes an XOR operation. From a computational perspective, single instances of hash operation and encryption operation have been considered equal [1].

Table 7.5: Overheads associated with PPKA protocol 1.

Index	**Node** N	**Hub Node** HN
Computation Overhead	$5h + 9\oplus$	$7h + 14\oplus$
Communication Overhead	$5B + 16$ bits	$4B + 16$ bits
Storage Overhead	$3B$ bits	B bits

Table 7.6: Overheads associated with PPKA protocol 2.

Index	**Node** N	**Hub Node** HN
Computation Overhead	$6h + 9\oplus$	$10h + 14\oplus$
Communication Overhead	$5B + 16$ bits	$5B + 16$ bits
Storage Overhead	$4B$ bits	B bits

7.7 Chapter Summary

We proposed two authenticated key agreement protocols suitable for WBANs. The protocols are based upon symmetric cryptographic components only and are thus highly efficient and avoid the additional burden of deploying and managing an associated PKI. Our protocols are suitable for any application scenario where efficiency is of essence and the network can be initialized by a "System Administrator". In addition to the requisite security guarantees, the proposed protocols also offer appropriate privacy attributes suitable for a wide variety of application scenarios. In order to ensure confidence in our proposals, we introduce formal security frameworks for the analysis of privacy-preserving key agreement protocols, and analyze our constructions. The proposed protocols emerge as attractive alternatives to the current key exchange methods described in the IEEE 802.15.6 standard, which are based upon legacy public-key based primitives and do not offer any privacy features.

Chapter 8

Conclusion

This chapter provides concluding remarks about the thesis.

8.1 Contributions Summary

Given recent progress towards achieving practical quantum computers [119], it is crucial that proposals for any cyber security standard should also cater for this potent future threat. In this regard, NIST's *Post-Quantum Cryptography Standardization* project [112] is an ongoing effort to standardize quantum-secure public-key cryptographic algorithms. The final draft standards are expected to be made publicly available by 2024. However, it is not clear whether the candidate encryption and signature schemes currently being evaluated by NIST will ultimately be suitable for deployment in constrained communication environments such as IoT, embedded SIMs, implantable wireless sensors, etc.

This thesis suggests that, instead of waiting for the outcome of the NIST standardization project, wherever possible, we should consider whether public-key cryptography can be substituted by symmetric cryptography in current security standards. This especially applies to scenarios where *a priori* relationships already exists between the protocol participants. Compared to quantum-secure public-key cryptography, such an approach offers benefits from a viability and timeliness viewpoint. Moreover, given the penetration of these modern communication technologies in our everyday lives, it is desirable that provisions improving user privacy also be made part of these standards. Unfortunately, this seems to be missing in many widely-adopted communication standards.

This thesis has explored and analyzed end-user privacy in one of the most important future communication standards, 5G (Chapter 3). We analyzed the current status of subscription privacy on the 5G radio interface and found that there are significant

issues which need rectifying. To shed light on this matter, we catalogued the privacy vulnerabilities that already exist in previous mobile telephony standards. The privacy improvements offered by the recently finalized 5G standard were identified. Consequently, we were able to highlight privacy issues from previous generations that remain unresolved in 5G Release 15. For completeness, we also explored new privacy attacks which surfaced after the publication of the 5G standard (Chapter 4). To address the identified privacy gaps, we also presented improvements.

This thesis presented a symmetric alternative to the current public-key based user identification scheme of the 5G standard (Chapter 5). As the current 5G identification scheme is the only public-key cryptography based mechanism within the standard, adoption of our alternative proposal will make the 5G standard independent of any quantum-vulnerable cryptography. We also developed a novel security framework titled Symmetric Updatable Private Authentication (SUPA) and provided a detailed formal analysis of our proposed scheme in this framework. Another contribution of this thesis was to combine our alternative 5G identification scheme with a downgrade protection proposal to come with a 5G identification mechanism which is both quantum-secure and downgrade-resistant (Chapter 6).

We adapted the techniques which we developed for our private 5G identification to come up with two efficient Privacy Preserving Key Agreement (PPKA) protocols for IEEE Std 802.15.6 (Chapter 7). Our PPKA protocols avoided any public-key cryptography and are thus well-suited to post-quantum deployment scenarios. We provided concise game-based security and privacy definitions capturing anonymity and unlinkability and proved formally that our key agreement protocols achieved these notions.

8.2 On Interaction with the Standardization Bodies

The journey from a novel cryptographic solution to a fully-specified and implementation-ready standard is like a bumpy ride down a long, winding road. We decided to embark on this journey and contacted the relevant standardization bodies' representatives about our work. The response was not particularly encouraging. As the organizational structure and rules of business of 3GPP's Technical Specification Groups (TSGs) and IEEE's Working Groups (WGs) are poles apart, we provide the details of our interaction with each one of them separately.

3GPP is an international industry consortia driven by commercial interests. One cannot participate in the 3GPP activities without a membership. We learned this the hard way by trying to contact the 3GPP SA3 chair via email. The reply never came. Fortunately, Royal Holloway's Information Security Group (ISG) has an ongoing collaboration with Vodafone, one of the 3GPP members. Steve Babbage of Vodafone was kind enough to advise that any proposal for 3GPP needs to tabled at one of its meetings by a member. We forwarded the details of our alternative SUPI protection proposal (Chapter 5) to Vodafone. Vodafone promised to look at it but warned that it might not be a 3GPP priority as they were already tied up due to commercial launching of 5G. Quantum-secure cryptography for 5G is still in a study phase [21] and may get looked at in Release 17 onwards.

For IEEE Std 802.15.6, we started by emailing the IEEE 802.15 WG chair regarding our proposals. An intermittent email interaction with the WG chair, which lasted from August, 2017 until March, 2019, started with the sending of a document summarizing our proposals. This document was to be distributed among the WG email group. The aim was to distribute "the idea" to the concerned audience for their opinion and interest and then decide upon any further course of action based on the response. Unfortunately, this response never came. All this explains why remedial measures have never been taken for serious security vulnerabilities discovered back in 2015 [134] in an "international" standard. Our conclusion from this experience is that the IEEE committees are, essentially, voluntary organizations and the change process involves a lot of bureaucratic hurdles. As a result, new standards do not develop quickly and major security modifications may take years.

8.3 Future Research Directions

The study concerning subscription privacy in 5G (Chapter 3) takes into account enhancements offered up to 3GPP Release 15 (5G Phase 1). The ongoing Release 16 (5G Phase 2) is planned to be placed into "frozen" status in March, 2020 with a target completion date of June, 2020 [143]. As the list of new features of Release 16 contains support for IoT devices, V2X and a new user identity paradigm, it will be interesting to evaluate the effect of these new features upon the current status of subscription privacy in 5G.

We also suggest to conducting a security analysis of the combined 5G-AKA protocol and our PQID proposal (Chapter 5) in an AKA security model. Further, our

SUPA security framework can be augmented to capture quantum adversaries, to show quantum security of our PQID scheme. Proving quantum security in such an enhanced model usually requires the proof strategy of *history-free reduction* [37].

The PPKA-2 protocol (Chapter 7) offers the security properties of Partial Forward Secrecy and KCI resilience for the WBAN standard IEEE Std 802.15.6 in case of compromise of the long-term secret (id_N) of the sensor/client node. It would be interesting to investigate whether future research can yield another PPKA protocol based on symmetric primitives and still offers (full) Forward Secrecy and KCI resilience in the (additional) event of compromise of the long-term secret (k_{HN}) of the Hub node.

Bibliography

[1] Crypto++ 5.6.5 Benchmarks. https://www.cryptopp.com/benchmarks.html. [Online; accessed 25-August-2017].

[2] FBI-Apple encryption dispute. https://en.wikipedia.org/wiki/FBI%E2%80%93Apple_encryption_dispute. [Online; accessed 11-March-2019].

[3] IEEE Standard for Local and Metropolitan Area Networks - Part 15.6: Wireless Body Area Networks. *IEEE Std 802.15.6-2012*, pages 1–271, Feb 2012. doi: 10.1109/IEEESTD.2012.6161600.

[4] *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019.

[5] 3rd Generation Partnership Project. Formal Analysis of the 3G Authentication Protocol 3GPP TR 33.902 Version 4.0.0 (Release 4), Sep 2001.

[6] 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; International Mobile station Equipment Identities (IMEI) (3GPP TS 22.016 Version 8.0.0 Release 08), Dec 2008.

[7] 3rd Generation Partnership Project. Rationale and track of security decisions in Long Term Evolution (LTE) RAN / 3GPP System Architecture Evolution (SAE) (3GPP TR 33.821 Version 9.0.0 Release 9), Jun 2009.

[8] 3rd Generation Partnership Project. Study on the security aspects of the next generation system (3GPP TR 33.899 Version 1.3.0 Release 14), Aug 2017.

[9] 3rd Generation Partnership Project. 3G Security; Security Architecture (3GPP TS 33.102 Version 15.0.0 Release 15), Jun 2018.

[10] 3rd Generation Partnership Project. 3G Security;Lawful Interception requirements (3GPP TS 33.106 Version 15.1.0 Release 15), Jun 2018.

[11] 3rd Generation Partnership Project. Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)(3GPP TS 33.220 Version 15.2.0 Release 15), June 2018.

[12] 3rd Generation Partnership Project. Mobile Application Part (MAP)Specification (3GPP TS 29.002 Version 15.3.0 Release 15), Mar 2018.

[13] 3rd Generation Partnership Project. Radio Resource Control (RRC); Protocol specification (3GPP TS 25.331 Version 15.4.0 Release 15), Sep 2018.

[14] 3rd Generation Partnership Project. Security Architecture and Procedures for 5G Systems (3GPP TS 33.501 Version 15.0.0 Release 15), Mar 2018.

[15] 3rd Generation Partnership Project. System Architecture for the 5G System (3GPP TS 23.501 Version 15.1.0 Release 15), Mar 2018.

[16] 3rd Generation Partnership Project. 3GPP System Architecture Evolution (SAE); Security architecture 3GPP TS 33.401 Version 15.8.0 (Release 15), June 2019.

[17] 3rd Generation Partnership Project. Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (3GPP TS 36.331 Version 15.6.0 Release 15), Jun 2019.

[18] 3rd Generation Partnership Project. NG-RAN; NG Application Protocol (NGAP)(3GPP TS 38.413 Version 15.3.0 Release 15), Mar 2019.

[19] 3rd Generation Partnership Project. NR; User Equipment (UE) procedures in Idle mode and RRC Inactive state (3GPP TS 38.304 Version 15.5.0 Release 15), Sep 2019.

[20] 3rd Generation Partnership Project. NR;Radio Resource Control (RRC) protocol specification (3GPP TS 38.331 Version 15.6.0 Release 15), Jun 2019.

[21] 3rd Generation Partnership Project. Study on the support of 256-bit algorithms for 5G (3GPP TR 33.841 Version 16.1.0 Release 16), Mar 2019.

[22] 3rd Generation Partnership Project. Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3 (Release 16) (3GPP TS 24.301 Version 16.2.0 Release 16), Sep 2019.

[23] 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; Study on 5G Security Enhancement against False Base Stations Version 0.6.0 (Release 16), Aug 2019.

[24] 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; Lawful Interception (LI) architecture and functions (3GPP TS 33.127 Version 16.4.0 Release 16), Jul 2020.

[25] 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; Lawful Interception requirements (3GPP TS 33.126 Version 16.2.0 Release 16), Jul 2020.

[26] 3rd Generation Partnership Project. Technical Specification Group Services and System Aspects; Protocol and procedures for Lawful Interception (LI); Stage 3 (3GPP TS 33.128 Version 16.3.0 Release 16), Jul 2020.

[27] Omer H. Abdelrahman and Erol Gelenbe. Signalling Storms in 3G Mobile Networks. In *IEEE International Conference on Communications, ICC 2014, Sydney, Australia, June 10-14, 2014*, pages 1017–1022. IEEE, 2014.

[28] Ijaz Ahmad, Tanesh Kumar, Madhusanka Liyanage, Jude Okwuibe, Mika Ylianttila, and Andrei V. Gurtov. 5G security: Analysis of threats and solutions. In *IEEE Conference on Standards for Communications and Networking, CSCN 2017, Helsinki, Finland, September 18-20, 2017*, pages 193–199. IEEE, 2017.

[29] Ijaz Ahmad, Tanesh Kumar, Madhusanka Liyanage, Jude Okwuibe, Mika Ylianttila, and Andrei V. Gurtov. Overview of 5G Security Challenges and Solutions. *IEEE Communications Standards Magazine*, 2(1):36–43, 2018.

[30] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, and Mark Ryan. Privacy through Pseudonymity in Mobile Telephony Systems. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*. The Internet Society, 2014.

[31] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. New Privacy Issues in Mobile Telephony: Fix and Verification. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 205–216. ACM, 2012.

[32] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, and Mark Dermot Ryan. Analysis of Privacy in Mobile Telephony Systems. *Int. J. Inf. Sec.*, 16(5):491–523, 2017.

[33] David A. Basin, Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, Ralf Sasse, and Vincent Stettler. A Formal Analysis of 5G Authentication. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 1383–1396. ACM, 2018.

[34] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249. Springer, 1993.

[35] Bruno Blanchet. Automatic Verification of Security Protocols in the Symbolic Model: The Verifier ProVerif. In Alessandro Aldini, Javier López, and Fabio Martinelli, editors, *Foundations of Security Analysis and Design VII - FOSAD 2012/2013 Tutorial Lectures*, volume 8604 of *Lecture Notes in Computer Science*, pages 54–87. Springer, 2013.

[36] Michael Bock. Simulation chamber and method for setting off explosive charges contained in freight in a controlled manner, May 2016. US Patent No. 9335139, Filed September 19th., 2012, Issued May. 10th., 2016.

[37] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random Oracles in a Quantum World. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.

[38] Ravishankar Borgaonkar, Lucca Hirschi, Shinjo Park, and Altaf Shaik. New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols. *PoPETs*, 2019(3):108–127, 2019.

[39] Ravishankar Borgaonkar, Lucca Hirshi, Shinjo Park, Altaf Shaik, Andrew Martin, and Jean-Pierre Seifert. New Adventures in Spying 3G & 4G Users: Locate, Track, Monitor. In *Blackhat, Las Vegas, USA 2017*, July 2017.

[40] Colin Boyd, Yvonne Cliff, Juan Manuel González Nieto, and Kenneth G. Paterson. Efficient one-round key exchange in the standard model. In Yi Mu, Willy Susilo, and Jennifer Seberry, editors, *Information Security and Privacy, 13th Australasian Conference, ACISP 2008, Wollongong, Australia, July 7-9, 2008, Proceedings*, volume 5107 of *Lecture Notes in Computer Science*, pages 69–83. Springer, 2008.

[41] Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Information Security and Cryptography. Springer, 2003.

[42] James J Caffery and Gordon L Stuber. Overview of radiolocation in CDMA cellular systems. *IEEE Communications Magazine*, 36(4):38–45, 1998.

[43] CATT. Solution for SUPI privacy and LI requirement. https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_90Bis_SanDiego/Docs/S3-180591.zip, Mar 2019.

[44] Riccardo Cavallari, Flavia Martelli, Ramona Rosini, Chiara Buratti, and Roberto Verdone. A survey on wireless body area networks: Technologies and design challenges. *IEEE Communications Surveys and Tutorials*, 16(3):1635–1657, 2014.

[45] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.

[46] Liqun Chen and Caroline Kudla. Identity based authenticated key agreement protocols from pairings. In *16th IEEE Computer Security Foundations Workshop (CSFW-16 2003), 30 June - 2 July 2003, Pacific Grove, CA, USA*, pages 219–233. IEEE Computer Society, 2003.

[47] Hung-Yu Chien. Authenticated diffie-hellman key agreement scheme that protects client anonymity and achieves half-forward secrecy. *Mobile Information Systems*, 2015:354586:1–354586:7, 2015.

[48] Gaurav Choudhary and Vishal Sharma. A Survey on the Security and the Evolution of Osmotic and Catalytic Computing for 5G Networks. *CoRR*, abs/1909.08844, 2019.

[49] Cas Cremers and Martin Dehnel-Wild. Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019* [4].

[50] Cas Cremers and Anja Lehmann, editors. *Security Standardisation Research - 4th International Conference, SSR 2018, Darmstadt, Germany, November 26-27, 2018, Proceedings*, volume 11322 of *Lecture Notes in Computer Science*. Springer, 2018.

[51] Neil J. Croft. On forensics: A silent SMS attack. In Hein S. Venter, Marianne Loock, and Marijke Coetzee, editors, *2012 Information Security for South Africa, Balalaika Hotel, Sandton, Johannesburg, South Africa, August 15-17, 2012*, pages 1–4. IEEE, 2012.

[52] Adrian Dabrowski, Georg Petzl, and Edgar R. Weippl. The Messenger Shoots Back: Network Operator Based IMSI Catcher Detection. In Fabian Monrose, Marc Dacier, Gregory Blanc, and Joaquín García-Alfaro, editors, *Research in Attacks, Intrusions, and Defenses - 19th International Symposium, RAID 2016, Paris, France, September 19-21, 2016, Proceedings*, volume 9854 of *Lecture Notes in Computer Science*, pages 279–302. Springer, 2016.

[53] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar R. Weippl. IMSI-catch me if you can: IMSI-catcher-catchers. In Charles N. Payne Jr., Adam Hahn, Kevin R. B. Butler, and Micah Sherr, editors, *Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, December 8-12, 2014*, pages 246–255. ACM, 2014.

[54] K. S. Deepak and A. V. Babu. Energy efficiency analysis of IEEE 802.15.6 based wireless body area networks in scheduled access mode. *Wireless Networks*, 22(5):1441–1459, 2016.

[55] Whitfield Diffie, Paul C. van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Des. Codes Cryptography*, 2(2):107–125, 1992.

[56] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. *IEEE Trans. Information Theory*, 29(2):198–207, 1983.

[57] Fred Donovan. Healthcare Data Breach Costs Remain Highest Among Industries. https://healthitsecurity.com/news/healthcare-data-breach-costs-remain-highest-among-industries, Jul 2018.

[58] Ericsson, Qualcomm Incorporated, Samsung, Huawei, Hisilicon, and Intel. SUCI and LI - verification hash integrated in 5G AKA. https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_90Bis_SanDiego/Docs/S3-180818.zip, Mar 2019.

[59] ETSI-SAGE. First response on ECIES for concealing IMSI or SUPI. https://portal.3gpp.org/ngppapp/CreateTdoc.aspx?mode=view&contributionId=832160, Oct 2017.

[60] Mohamed Amine Ferrag, Leandros A. Maglaras, Antonios Argyriou, Dimitrios Kosmanos, and Helge Janicke. Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes. *J. Network and Computer Applications*, 101:55–82, 2018.

[61] Dan Forsberg, Leping Huang, Tsuyoshi Kashima, and Seppo Alanärä. Enhancing Security and Privacy in 3GPP E-UTRAN Radio Interface. In *Proceedings of the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2007, 3-7 September 2007, Athens, Greece*, pages 1–5. IEEE, 2007.

[62] Pierre-Alain Fouque, Cristina Onete, and Benjamin Richard. Achieving Better Privacy for the 3GPP AKA Protocol. *PoPETs*, 2016(4):255–275, 2016.

[63] Dirk Fox. Der imsi-catcher. *Datenschutz und Datensicherheit*, 26(4), 2002.

[64] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[65] Pimmy Gandotra and Rakesh Kumar Jha. A survey on green communication and security challenges in 5G wireless communication networks. *J. Network and Computer Applications*, 96:39–61, 2017.

[66] Stuart Owen Goldman, Richard E Krock, Karl F Rauscher, and James Philip Runyon. Mobile forced premature detonation of improvised explosive devices via wireless phone signaling, Jun 2009. US Patent No. 7552670, Filed September 22nd., 2005, Issued Jun. 30th., 2009.

[67] Glen Greenwald. NSA collecting phone records of millions of Verizon customers daily. https://www.theguardian.com/world/2013/jun/06/nsa-phone-records-verizon-court-order, Jun 2013. [Online; accessed 23-September-2019].

[68] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996.

[69] Darrel Hankerson and Alfred Menezes. Elliptic Curve Cryptography. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security, 2nd Ed.*, page 397. Springer, 2011.

[70] Thaier Hayajneh, Ghada A. Al-Mashaqbeh, Sana Ullah, and Athanasios V. Vasilakos. A survey of wireless technologies coexistence in WBAN: analysis and open research issues. *Wireless Networks*, 20(8):2165–2199, 2014.

[71] Debiao He, Sherali Zeadally, Neeraj Kumar, and Jong-Hyouk Lee. Anonymous authentication for wireless body area networks with provable security. *IEEE Systems Journal*, PP(99):1–12, 2016.

[72] Byeongdo Hong, Sangwook Bae, and Yongdae Kim. GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018.

[73] Syed Rafiul Hussain, Mitziu Echeverria, Omar Chowdhury, Ninghui Li, and Elisa Bertino. Privacy attacks to the 4g and 5g cellular paging protocols using side channel information. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019* [4].

[74] Nathaniel Husted and Steven Myers. Mobile Location Tracking in Metro Areas: Malnets and Others. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 85–96. ACM, 2010.

[75] Markus Jakobsson and Susanne Wetzel. Security Weaknesses in Bluetooth. In David Naccache, editor, *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, volume 2020 of *Lecture Notes in Computer Science*, pages 176–191. Springer, 2001.

[76] Qi Jiang, Xinxin Lian, Chao Yang, Jianfeng Ma, Youliang Tian, and Yuanyuan Yang. A bilinear pairing based anonymous authentication scheme in wireless body area networks for mhealth. *J. Medical Systems*, 40(11):231:1–231:10, 2016.

[77] Roger Piqueras Jover. LTE security, protocol exploits and location tracking experimentation with low-cost software radio. *CoRR*, abs/1607.05171, 2016.

[78] Roger Piqueras Jover. The current state of affairs in 5G security and the main remaining security challenges. *CoRR*, abs/1904.08394, 2019.

[79] Roger Piqueras Jover and Vuk Marojevic. Security and Protocol Exploit Analysis of the 5G Specifications. *IEEE Access*, 7:24956–24963, 2019.

[80] Julian Kelly. A Preview of Bristlecone, Google's New Quantum Processor. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html. [Online; accessed 08-June-2018].

[81] Haibat Khan. An identity based routing path verification scheme for wireless sensor networks. *IJSNet*, 26(1):54–68, 2018.

[82] Haibat Khan, Benjamin Dowling, and Keith M. Martin. Highly efficient privacy-preserving key agreement for wireless body area networks. In *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 12th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2018, New York, NY, USA, August 1-3, 2018*, pages 1064–1069. IEEE, 2018.

[83] Haibat Khan, Benjamin Dowling, and Keith M. Martin. Identity Confidentiality in 5G Mobile Telephony Systems. In Cremers and Lehmann [50], pages 120–142.

[84] Haibat Khan and Keith M. Martin. On the Efficacy of New Privacy Attacks against 5G AKA. In Mohammad S. Obaidat and Pierangela Samarati, editors, *Proceedings of the 16th International Joint Conference on e-Business and*

Telecommunications, ICETE 2019 - Volume 2: SECRYPT, Prague, Czech Republic, July 26-28, 2019., pages 431–438. SciTePress, 2019.

[85] Haibat Khan and Keith M. Martin. A Survey of Subscription Privacy on the 5G Radio Interface - The Past, Present and Future. *J. Inf. Secur. Appl.*, 53:1–17, 2020.

[86] Jamil Y. Khan, Mehmet R. Yuce, Garrick Bulger, and Benjamin Harding. Wireless body area network (WBAN) design techniques and performance evaluation. *J. Medical Systems*, 36(3):1441–1457, 2012.

[87] Mohammed Shafiul Alam Khan and Chris J. Mitchell. Improving air interface user privacy in mobile telephony. In Liqun Chen and Shin'ichiro Matsuo, editors, *Security Standardisation Research - Second International Conference, SSR 2015, Tokyo, Japan, December 15-16, 2015, Proceedings*, volume 9497 of *Lecture Notes in Computer Science*, pages 165–184. Springer, 2015.

[88] Mohammed Shafiul Alam Khan and Chris J. Mitchell. Trashing IMSI catchers in mobile networks. In Guevara Noubir, Mauro Conti, and Sneha Kumar Kasera, editors, *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, Boston, MA, USA, July 18-20, 2017*, pages 207–218. ACM, 2017.

[89] Mohsin Khan, Philip Ginzboorg, Kimmo Järvinen, and Valtteri Niemi. Defeating the Downgrade Attack on Identity Privacy in 5G. In Cremers and Lehmann [50], pages 95–119.

[90] Mohsin Khan, Kimmo Järvinen, Philip Ginzboorg, and Valtteri Niemi. On Desynchronization of User Pseudonyms in Mobile Networks. In Rudrapatna K. Shyamasundar, Virendra Singh, and Jaideep Vaidya, editors, *Information Systems Security - 13th International Conference, ICISS 2017, Mumbai, India, December 16-20, 2017, Proceedings*, volume 10717 of *Lecture Notes in Computer Science*, pages 347–366. Springer, 2017.

[91] Mohsin Khan and Valtteri Niemi. Concealing IMSI in 5G Network Using Identity Based Encryption. In Zheng Yan, Refik Molva, Wojciech Mazurczyk, and Raimo Kantola, editors, *Network and System Security - 11th International Conference, NSS 2017, Helsinki, Finland, August 21-23, 2017, Proceedings*, volume 10394 of *Lecture Notes in Computer Science*, pages 544–554. Springer, 2017.

[92] Mohsin Khan, Valtteri Niemi, and Philip Ginzboorg. IMSI-based Routing and Identity Privacy in 5G. In *Proceedings of the 22nd Conference of Open Innovations Association FRUCT, Jyvaskyla, Finland*, 2018.

[93] Rabia Khan, Pardeep Kumar, Dushantha Nalin K Jayakody, and Madhusanka Liyanage. A survey on security and privacy of 5G technologies: Potential solutions, recent advancements and future directions. *IEEE Communications Surveys & Tutorials*, 2019.

[94] Adrien Koutsos. The 5g-aka authentication protocol privacy. In *IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019*, pages 464–479. IEEE, 2019.

[95] KPN, NTT DOCOMO, DT, BT, and NEC. Proposal and Discussion for Privacy and LI Solution. https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_90Bis_SanDiego/Docs/S3-180684.zip, Mar 2019.

[96] Hugo Krawczyk. HMQV: A high-performance secure diffie-hellman protocol. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 546–566. Springer, 2005.

[97] DF Kune, J Koelndorfer, N Hopper, and Y Kim. Location Leaks on the GSM Air Interface. In *19th Annual Network & Distributed System Security Symposium, ISOC-NDSS*, 2012.

[98] Andreas Kunz and Xiaowei Zhang. New 3GPP Security Features in 5G Phase 1. In *2018 IEEE Conference on Standards for Communications and Networking, CSCN 2018, Paris, France, October 29-31, 2018*, pages 1–6. IEEE, 2018.

[99] Ben Lampert, Riad S. Wahby, Shane Leonard, and Philip Levis. Robust, low-cost, auditable random number generation for embedded system security. In Philip Levis, Steve Eglash, Lama Nachman, and Anthony Rowe, editors, *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems, SenSys 2016, Stanford, CA, USA, November 14-16, 2016*, pages 16–27. ACM, 2016.

[100] Laurie Law, Alfred Menezes, Minghua Qu, Jerome A. Solinas, and Scott A. Vanstone. An efficient protocol for authenticated key agreement. *Des. Codes Cryptography*, 28(2):119–134, 2003.

[101] Ming-Feng Lee, Nigel P. Smart, Bogdan Warinschi, and Gaven J. Watson. Anonymity guarantees of the UMTS/LTE authentication and connection protocol. *Int. J. Inf. Sec.*, 13(6):513–527, 2014.

[102] Ming Li, Wenjing Lou, and Kui Ren. Data security and privacy in wireless body area networks. *IEEE Wireless Commun.*, 17(1):51–58, 2010.

[103] Xiong Li, Maged Hamada Ibrahim, Saru Kumari, Arun Kumar Sangaiah, Vidushi Gupta, and Kim-Kwang Raymond Choo. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. *Computer Networks*, 129:429–443, 2017.

[104] Andy Lilly. IMSI catchers: hacking mobile communications. *Network Security*, 2017(2):5–7, 2017.

[105] John Mattsson. Post-quantum cryptography in mobile networks. 2017.

[106] Simon Meier, Benedikt Schmidt, Cas Cremers, and David A. Basin. The TAMARIN Prover for the Symbolic Analysis of Security Protocols. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, pages 696–701. Springer, 2013.

[107] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[108] Stig Fr. Mjølsnes and Ruxandra F. Olimid. Easy 4G/LTE IMSI Catchers for Non-Programmers. In Jacek Rak, John Bay, Igor V. Kotenko, Leonard J. Popyack, Victor A. Skormin, and Krzysztof Szczypiorski, editors, *Computer Network Security - 7th International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2017, Warsaw, Poland, August 28-30, 2017, Proceedings*, volume 10446 of *Lecture Notes in Computer Science*, pages 235–246. Springer, 2017.

[109] Stig Fr. Mjølsnes and Ruxandra F. Olimid. Private Identification of Subscribers in Mobile Networks: Status and Challenges. *IEEE Communications Magazine*, 57(9):138–144, 2019.

[110] Samaneh Movassaghi, Mehran Abolhasan, Justin Lipman, David B. Smith, and Abbas Jamalipour. Wireless body area networks: A survey. *IEEE Communications Surveys and Tutorials*, 16(3):1658–1686, 2014.

[111] United Nations. Ageing. https://www.un.org/en/sections/issues-depth/ageing/, Jan 2020.

[112] NIST. Post-Quantum Cryptography Standardization. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization, Jan 2020.

[113] Karsten Nohl. Mobile Self-defense. In *31st Chaos Communication Congress 31C3*, 2014.

[114] Karsten Nohl and Sylvain Munaut. Wideband GSM Sniffing. In *27th Chaos Communication Conference*, 2010.

[115] Nokia. Discussion on LI conformity by verification hash method. https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_90Bis_SanDiego/Docs/S3-180768.zip, Mar 2019.

[116] Nokia, Gemalto, and IDEMIA. SUCI and LI verification hash integrated in 5G AKA. https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_90Bis_SanDiego/Docs/S3-180769.zip, Mar 2019.

[117] U.S. Department of Health & Human Services. The HIPAA Privacy Rule. https://www.hhs.gov/hipaa/for-professionals/privacy/index.html, Aug 2002.

[118] Ruxandra F Olimid and Stig F Mjølsnes. On Low-Cost Privacy Exposure Attacks in LTE Mobile Communication. *Proceedings of the Romanian Academy Series A-Mathematics Physics Technical Sciences Information Science*, 18:361–370, 2017.

[119] Dennis Overbye. Quantum Computing Is Coming, Bit by Qubit. https://www.nytimes.com/2019/10/21/science/quantum-computer-physics-qubits.html, Oct 2019.

[120] Chris Paget. Practical Cellphone Spying. *Def Con*, 18, 2010.

[121] DongGook Park, Colin Boyd, and Sang-Jae Moon. Forward secrecy and its application to future mobile communications security. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography, Third International Workshop on Practice and Theory in Public Key Cryptography, PKC 2000, Melbourne, Victoria, Australia, January 18-20, 2000, Proceedings*, volume 1751 of *Lecture Notes in Computer Science*, pages 433–445. Springer, 2000.

[122] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. 2010.

[123] Anand R Prasad, Sivabalan Arumugam, B Sheeba, and Alf Zugenmaier. 3GPP 5G Security. *Journal of ICT Standardization*, 6(1):137–158, 2018.

[124] Anil Kumar Rangisetti and Bheemarjuna Reddy Tamma. Software Defined Wireless Networks: A Survey of Issues and Solutions. *Wireless Personal Communications*, 97(4):6019–6053, 2017.

[125] Ravishankar Ravindran, Asit Chakraborti, Syed Obaid Amin, Aytac Azgin, and Guoqiang Wang. 5g-icn: Delivering ICN services over 5g using network slicing. *IEEE Communications Magazine*, 55(5):101–107, 2017.

[126] David Rupprecht, Adrian Dabrowski, Thorsten Holz, Edgar R. Weippl, and Christina Pöpper. On security research towards future mobile network generations. *IEEE Communications Surveys and Tutorials*, 20(3):2518–2542, 2018.

[127] Paul Schrodt. Edward Snowden just made an impassioned argument for why privacy is the most important right. https://www.businessinsider.com/edward-snowden-privacy-argument-2016-9?r=US&IR=T, Sep 2016.

[128] SECG SEC 1. Recommended Elliptic Curve Cryptography, Version 2.0. http://www.secg.org/sec1-v2.pdf, 2009.

[129] Altaf Shaik, Jean-Pierre Seifert, Ravishankar Borgaonkar, N. Asokan, and Valtteri Niemi. Practical attacks against privacy and availability in 4g/lte mobile communication systems. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society, 2016.

[130] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society, 1994.

[131] Victor Shoup. A proposal for an ISO standard for public key encryption. *IACR Cryptology ePrint Archive*, 2001:112, 2001.

[132] Christian Sørseth, Shelley Xianyu Zhou, Stig F Mjølsnes, and Ruxandra F Olimid. Experimental Analysis of Subscribers' Privacy Exposure by LTE Paging. *Wireless Personal Communications*, pages 1–19, 2018.

[133] Qiang Tang. *Key Establishment Protocols and Timed-Release Encryption Schemes*. PhD thesis, Department of Mathematics, Royal Holloway, University of London, Royal Holloway Research Online, 10 2007.

[134] Mohsen Toorani. On vulnerabilities of the security association in the IEEE 802.15.6 standard. In Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff, editors, *Financial Cryptography and Data Security - FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*, volume 8976 of *Lecture Notes in Computer Science*, pages 245–260. Springer, 2015.

[135] Reza Tourani, Satyajayant Misra, Travis Mick, and Gaurav Panwar. Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *IEEE Communications Surveys and Tutorials*, 20(1):566–600, 2018.

[136] Sana Ullah, Henry Higgins, Bart Braem, Benoît Latré, Chris Blondia, Ingrid Moerman, Shahnaz Saleem, Ziaur Rahman, and Kyung Sup Kwak. A comprehensive survey of wireless body area networks - on phy, mac, and network layers solutions. *J. Medical Systems*, 36(3):1065–1094, 2012.

[137] European Union. Regulation (EU) 2016/679 (General Data Protection Regulation). https://gdpr-info.eu/, May 2016.

[138] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. Defeating IMSI catchers. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 340–351. ACM, 2015.

[139] Henk C. A. van Tilborg and Sushil Jajodia, editors. *Encyclopedia of Cryptography and Security, 2nd Ed.* Springer, 2011.

[140] Vodafone. Discussion paper on embedded routing information in SUCI. https://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_90Bis_SanDiego/docs/S3-180761.zip, Mar 2019.

[141] Vodafone. pCR to 33.501 - addition of routing information into SUCI. https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_90Bis_SanDiego/Docs/S3-180763.zip, Mar 2019.

[142] Chunzhi Wang and Yanmei Zhang. New authentication scheme for wireless body area networks using the bilinear pairing. *J. Medical Systems*, 39(11):136:1–136:8, 2015.

[143] Alan Weissberger. 3GPP Release 16 Update: 5G Phase 2 (including URLLC) to be completed in June 2020; Mission Critical apps extended. https://techblog.comsoc.org/2019/10/06/3gpp-release-16-update-5g-phase-2-including-urllc-to-be-completed-in-june-2020/, Oct 2019.

[144] Masaya Yasuda, Takeshi Shimoyama, Jun Kogure, and Tetsuya Izu. Computational hardness of IFP and ECDLP. *Appl. Algebra Eng. Commun. Comput.*, 27(6):493–521, 2016.